1 453 607

PATENT SPECIFICATION

(11) 1 453 607

(21) Application No. 59038/73 (22) Filed 20 Dec. 1973
(31) Convention Application No. 318 451
(32) Filed 26 Dec. 1972 in
(33) United States of America (US)
(44) Complete Specification published 27 Oct. 1976
(51) INT CL² G06F 15/02
(52) Index at acceptance
G4A 10EX 11C 16D 17BX 17P 2F5 3W10W2 5A 5B 9C 9F 9X

(19)

(54) ADAPTABLE PROGRAMMED CALCULATOR

(71) We, HEWLETT-PACKARD COMPANY of 1501 Page Mill Road, Palo Alto, California 94304, United States of America, an American company organized and existing under the laws of the State of California, United States of America, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

This invention relates generally to calculators and improvements therein and more particularly to calculators which may be easily adapted to meet the specific needs of each user.

The present invention is closely related to that disclosed and claimed in the complete specifications of United Kingdom Patent Applications Nos. 50310/74, 50311/74 and 50312/74. (Serial Nos. 1453608, 1453609 and 1453610).

Calculators constructed according to the prior art have generally taken one or two approaches toward reducing the labor content of repetitious, routine computational tasks. The first is by means of programmability. The programmable machine has the inherent advantage of program versatility in that it can be programmed to solve problems encountered in nearly all disciplines including mathematics, science, engineering, business, finance, statistics, etc. Unfortunately, this versatility has added significantly to cost. The user is, therefore, paying for considerably more calculating capability and versatility than is required, for instance, in solving repetitive problems related to the same discipline. The approach which has recently been taken to solve this problem is that of a "dedicated" calculator. Such a machine generally has built-in fixed programming which allows it to handle only a narrow range of problems. Even though these dedicated calculators are less expensive than programmable types, they have a serious short-coming in that their programs can not be changed.

In the complete specification of the aforesaid application No. 50310/74 (Serial No. 1453608) there is disclosed and claimed an electronic calculator comprising: keyboard input means for entering information into the calculator, said keyboard input means including a plurality of command keys and a plurality of data keys, each command key and each data key being associated with a separated mnemonic code; main memory means for storing microprogrammed routines and subroutines performed by the calculator in executing selected keyboard commands; read-only memory means coupled to said main memory means for storing keyboard mnemonic codes representing one or more programs employing selected keyboard commands and data; processing means coupled to said keyboard input means, read-only memory means, and main memory means, said processing means being selectively responsive to actuation of a command key or to the mnemonic code associated with a command key, when that mnemonic code is encountered in processing a program stored in said read-only memory means, for executing the microprogrammed routines and subroutines stored in said main memory means that are associated with that command key or keyboard mnemonic code; and output means coupled to said processing means for providing an output indication of the results of selected commands executed by the calculator.

In the complete specification of the aforesaid application No. 50311/74 (Serial No. 1453609) there is disclosed and claimed an electronic calculator comprising: keyboard input means for entering information into the calculator; memory means for storing information entered into the calculator; processing means responsive to information entered from said keyboard input means and to operating states

within the calculator for selectively performing calculations and for providing an output indication of the results of those calculations; and logic means responsive to designation of an automatic decimal point mode for automatically placing a decimal point at any predetermined one of a plurality of positions in noninteger data subsequently entered into the calculator from said keyboard input means without the necessity of manual entry of the decimal point in the noninteger data by the user.

In the complete specification of the aforesaid application No. 50312/74 (Serial No. 1453610) there is disclosed and claimed an electronic calculator comprising: keyboard input means for entering information into the calculator, said keyboard input means including a percent key for entering a percent operator into the calculator and a plurality of keys for entering data and arithmetic operators including an addition operator into the calculator; and processing means coupled to said keyboard input means and responsive to information entered into the calculator from said keyboard input means and to operating states within the calculator for selectively performing calculations and for providing an output indication of the results of those calculations, said processing means being responsive to entry of a first number followed by entry of the addition operator followed by entry of a second number followed by entry of the percent operator for calculating the sum of the first number and a percentage thereof as specified by the second number.

The present invention provides an electronic calculator comprising: a keyboard input unit for providing the user with a plurality of keyboard functions and for entering into the calculator information used to initiate selected ones of those keyboard functions and data used in making selected calculations to perform the selected keyboard functions; a memory unit for storing routines and subroutines of instructions to be performed by the calculator in making the selected calculations to perform the selected keyboard functions; a modular keyboard section, including a read-only memory containing additional routines and subroutines of instructions, for providing the user with additional keyboard functions and for entering into the calculator information used to initiate selected ones of those additional keyboard functions; processing means responsive to information from the keyboard input unit and to operating states within the calculator itself for selectively performing one or more of the rountines and subroutines stored in the memory unit to make the selected calculations employing data entered from the keyboard input unit; said processing means being responsive to information from the modular keyboard section and to operating states within the calculator itself for selectively performing one or more of the additional routines and subroutines of instructions contained in the read-only memory to make selected calculations employing data entered from at least one of the keyboard input unit and the modular keyboard section; and output means for providing an output indication of the results of the calculations performed by the processing means.

In a calculator as set forth in the last preceding paragraph said keyboard input means may include a receptacle into which said modular keyboard section may be removably plugged; and said keyboard input means and said modular keyboard section may include electrical connection means for electrically connecting said keyboard input means and said modular keyboard section.

In a calculator as set forth in the last preceding paragraph or the last preceding paragraph but one said modular keyboard section may include a plurality of keys each of which represents a predetermined function. Alternatively said modular keyboard section includes a plurality of keys each of which may be associated with a function defined by the user.

A calculator as set forth in any one of the last three immediately preceding paragraphs may further comprise interface means for coupling a selected external input/output peripheral unit to the calculator, said interface means including logic means for transferring data between the calculator and the selected input/output peripheral unit and further including microprogrammed read-only memory means for enabling said processing means to perform logic operations associated with the selected input/output peripheral unit.

The keyboard input unit of a calculator according to any preferred embodiment of the present invention includes a group of data keys for entering numeric data into the calculator, a group of control keys for controlling the various modes of the calculator and the operations of the output printer, a group of operand keys for designating the mathematical operations to be performed on

various items of data, and a group of program keys for controlling the execution of library programs stored within a plug-in read-only memory (ROM) or programmable read-only memory (PROM).

The keyboard also includes a blank section which will accommodate a plug-in function block containing fifteen keys and associated read-only memory. Various function blocks may be dedicated to different disciplines and problems solving areas. For example, a dedicated function block oriented toward statistics includes keys whose representative functions would be helpful in solving statistical problems. Likewise, a mathematics function block would include various mathematical functions available as the result of key actuations. In addition, a user-definable function block may be plugged into the calculator keyboard. This block contains fifteen keys, each having a transparent cap which the user may remove for the purpose of inserting a function label. Each of these keys is associated with a particular function or program contained within a plug-in ROM or PROM currently employed with the calculator. Such function or program may then be called by simply actuating the associated key of the user-definable function block.

The optional 15-digit LED output display unit is contained within a plug-in printed circuit board which is automatically accommodated by the calculator.

The 18-column output printer unit is an integral part of the calculator and gives a printed record of entered data, arithmetic operators, calculated results, and diagnostic notes. Printing may be suppressed and otherwise controlled by means of keys on the keyboard input unit.

The MOS/LSI circuits include eight read-only memory circuits in which subroutines for performing various functions are stored. These circuits also supervise program execution and serve to control any peripheral input/output units which may be connected to the calculator. The read-only memory group comprises these eight individual read-only memory circuits which are identical in structure and differ only in the way in which they are programmed.

A control and timing circuit is used for scanning the keyboard, for retaining status information relating to the condition of the calculator or of a particular subroutine, and for generating a next address in read-only memory.

An arithmetic and register circuit contains an adder, a group of working registers, a group of data storage registers forming a stack, and a constant storage register for storing microprogramming level flags associated with various subroutines.

A data storage circuit provides ten data storage registers, five of which are used for parentheses nesting, three of which are used for performing various internal system housekeeping functions at the microprogram level, one of which is a grand total register which may be interrogated by the user, and the last of which is accessible to the user for storing a single item of data.

The last MOS/LSI circuit, the input/output (I/O) circuit, enables the calculator to communicate with various I/O peripheral units such as a typewriter or an X—Y plotter and determines whether the proper peripheral configuration for running a particular program is present. It also includes a binary arithmetic logic unit (ALU) for performing binary arithmetic, a program address counter used in running programs, and the necessary logic circuit for driving the internal printer unit.

The calculator may be operated manually from the keyboard input unit utilizing functions available as keys on the basic keyboard, on a dedicated plug-in function block, or on a user-definable plug-in function block. The calculator may also be operated automatically from a program comprising user-level language instructions and stored in a plug-in read-only memory unit (ROM), a plug-in programmable read-only memory unit (PROM) or a read/write memory unit associated with a plug-in magnetic card reading and recording unit.

There now follows a detailed description which is to be read with reference to the accompanying drawings of an electronic calculator according to the invention; it is to be clearly understood that this calculator has been selected for description to illustrate the invention by way of example and not by way of limitation.

In the drawings:—

Figure 1 is a front perspective view of an adaptable calculator according to the preferred embodiment of this invention;

Figure 2 is a rear perspective view of the adaptable calculator of Figure 1;

Figure 3 is a block diagram of the calculator of Figure 1;

Figure 4 is a detailed schematic diagram showing the interconnection of the arithmetic and register circuit and the control and logic circuit of Figure 3;

Figure 5 is a waveform diagram illustrating the timing sequence of the interconnecting busses of Figure 3;

Figure 6 is a block diagram of the control and timing circuit of Figure 3;

Figure 7 is a more detailed block diagram of the keyboard scanning circuitry of Figure 6;

Figure 8 is a detailed schematic diagram of the keyboard circuitry of Figure 7.

Figure 9 is a block diagram of the arithmetic and register circuit of Figure 3;

Figure 10 is a path diagram of the actual data paths for the register A—F and M of Figure 9;

Figure 11 is a waveform diagram illustrating the output signals for the display decoder outputs A—E of Figures 9 and 10;

Figure 12 is a waveform diagram illustrating the actual signals on the display decoder outputs A—E of Figures 9 and 10 when the digit 9 is decoded;

Figure 13 is a waveform diagram illustrating the timing of the START signal generated by the display decoder of Figure 9;

Figure 14 is a schematic diagram of the clock driver of Figure 4;

Figure 15 is a waveform diagram illustrating the timing relationship between the input and output signals of the clock driver of Figure 14;

Figure 16 is a detailed schematic diagram of the read-only memory group of Figure 3;

Figure 17 is a block diagram of one of the read-only memory circuits Ø–7 of Figure 16;

Figure 18 is a waveform illustrating a typical address signal and a typical instruction signal;

Figure 19 is a timing diagram illustrating the important timing points for a typical addressing sequence;

Figure 20 is a waveform diagram illustrating the word select signals generated in the control and timing circuit of Figures 3 and 6 and in the read-only memory circuits Ø–7 of Figure 3 and 17;

Figure 21 is a detailed schematic diagram of the input/output (I/O) processor of Figure 3;

Figures 22A—B are a block diagram of the I/O circuit of Figure 21;

Figure 23 is a waveform diagram illustrating the timing relationship between the SYNC, START, EERA, I_s, SRI, and IXT signals of the I/O circuit of Figure 22;

Figure 24 is a detailed schematic diagram of the data storage assembly of Figure 3;

Figures 25A—B are a block diagram of the data storage circuit of Figure 24;

Figure 26 is a waveform diagram illustrating the timing relationship between the SYNC, START, I_s, and BCD signals of the data storage circuit of Figure 25;

Figure 27 is a detailed schematic diagram of the timing circuitry associated with the printer of Figure 3;

Figure 28 is a detailed schematic diagram of the driver circuitry associated with the printer of Figure 3;

Figure 29 is a block diagram of the LED display of Figure 3;

Figure 30 is a logic diagram of the anode driver of Figure 29;

Figure 31 is a waveform diagram illustrating the timing relationship between various signals associated with the anode driver of Figures 29 and 30;

Figure 32 is a schematic diagram of the basic inductive drive circuit for one of the light emitting diodes employed in the LED display of Figures 3 and 29;

Figure 33 is a waveform diagram illustrating the timing relationship between the decimal point drive signals for the LED display of Figures 3 and 29;

Figure 34 is a schematic diagram of the inductive drive circuit for one digit of the LED display of Figures 3 and 29;

Figure 35 is a logic diagram of the cathode driver of Figure 29;

Figure 36 is a schematic diagram of the LED display of Figure 3;

Figure 37 is a schematic diagram of one segment of the LED display of Figures 3 and 36;

Figure 38 is an equivalent piecewise-linear model for the circuitry of Figure 37;

Figure 39 is a waveform diagram illustrating the inductor current and LED anode voltages associated with the circuitry of Figure 37;

Figure 40 is a schematic diagram of a system clock generator employed by the calculator in the absence of the LED display option;

Figure 41 is a schematic diagram of a power supply system which may be employed by the calculator of Figure 1;

Figure 42 is a flow chart of the overall microprograming system employed with the calculator of Figure 1;

Figure 43 is a flow chart of a display wait loop employed in the calculator of Figure 1;

Figure 44 is a detailed schematic diagram of the ROM group select circuitry contained within the calculator;

Figure 45 is a detailed schematic diagram of a plug-in keyboard function block which may be employed in the calculator of Figure 1;

Figure 46 is a detailed schematic diagram of an optional plug-in ROM/PROM which may be employed in the calculator of Figure 1;

Figure 47 is a detailed schematic diagram of an optional data storage unit which may be plugged into the calculator of Figure 1; and

Figure 48 is a flow chart showing the microprograming steps involved in an open parenthesis routine employed in the calculator.

SYSTEM ARCHITECTURE

Referring to Figures 1 and 2, there is shown a desk top electronic calculator 10 including a keyboard input unit 12 for entering data and instructions into the calculator, an optional seven-segment LED output display unit 14 for displaying each data entry and the results of calculations performed by the calculator. The calculator also includes an 18-column output printer unit 16 for printing intermediate and final results of calculations, entered data, arithmetic operators, and diagnostic notes. The keyboard input unit 12 also includes a covered blank section 18 which will accommodate a 15-key function block 20 for expanding the capabilities of the calculator. The function block 20 may be oriented toward a particular problem solving area in which case its keys will represent functions which are useful in making calculations relating to that discipline. Alternatively, the function block 20 may be a user-definable type in which the various keys may be labeled and defined to be functions or programs stored in a ROM or PROM 22 tailored to the requirements of each user. The function or program so defined may then be executed by simply actuating the associated key on the user-definable function block 20. The ROM or PROM 22 may be removably plugged into the calculator by means of a hinged top cover 24. The calculator may employ a ROM or PROM 22 without also employing a plug-in function block 20. In such a case, the ROM or PROM contains one or more programs which the user may execute from the main keyboard.

As shown in Figure 2, the calculator also includes three I/O receptacles 26 covered when not in use by receptacle caps 28. These receptacles serve to connect the calculator to various I/O peripheral units such as a typewriter, a marked sense card reader, an X—Y plotter, etc.

As shown in the overall block diagram of Figure 3, the calculator also includes as MOS/LSI control and timing circuit 30, and MOS/LSI arithmetic and register circuit 32, eight MOS/LSI read-only memory circuits comprising a ROM group 34, an MOS/LSI data storage circuit contained within a data storage assembly 36, and an MOS/LSI input/output circuit contained within an input/output processor 38.

The MOS/LSI circuits are two-phase dynamic types with low thresholds for assuring compatibility with standard TTL bipolar circuits and for operation at very low power levels. They are organized to process fourteen-digit BCD words in a digit-serial, bit-serial manner. They are also capable of bit-serially processing fifty-six-bit binary words. The maximum bit rate or clock frequency is two hundred kilohertz, which gives a word time of two hundred eighty microseconds. This means that a floating point addition may be accomplished in sixty milliseconds.

The control and timing circuit 30, the ROM group 34, the arithmetic and register circuit 32, the data storage assembly 36, and the input/output processor 38 are tied together by a six-line bus 40. This bus comprises a SYNC line, an instruction (I_s) line, a word select (WS) line, an instruction address (I_a) line, a START line, and a BCD line. All operations occur on a fifty-six-bit (b_0—b_{55}) word cycle (fourteen four-bit BCD digits). The timing sequence for some of the interconnecting lines comprises the bus 40 are shown in Figure 4.

The SYNC line carries synchronization signals from the control and timing circuit 30 to ROM circuits Ø–7 in the ROM group 34 and to the arithmetic and register circuit 32 to synchronize the calculator system. It provides one output each word time. This output also functions as a ten-bit wide window (b_{45}—b_{54}) during which the instruction line I_s is active.

The instruction line I_s carries ten-bit instructions from the active ROM circuit of the ROM group 34 to the other ROM circuits of the ROM group 34, to the control and timing circuit 30, to the data storage assembly 36, and to the input/output processor 38, each of which decodes the instructions locally and responds to or acts upon them if they pertain thereto and ignores them if they do not. In order to free instruction bit patterns normally associated with the arithmetic and register circuit and the control and timing circuit, the I_s line is gated prior to being examined by these circuits. For example, an ADD instruction affects the arithmetic and register circuit 32 but is ignored by all other circuits. Similarly, a SET STATUS BIT 5 instruction sets status flip-flop 5 in control and timing circuit 30 but is ignored by all other circuits.

The actual implementation of an instruction is delayed one word time from its receipt. For instance, an instruction may required the addition of digit 2 in two of the registers in the arithmetic and register circuit 32. The ADD instruction would be received by the arithmetic and register circuit 32 during bit times b_{45}—b_{54} of word time N+1. Thus, while one instruction is being executed, the next instruction is being fetched.

The WS line carries an enable signal from the control and timing circuit 30 or one of the ROM circuits in the ROM group 34 to the arithmetic and register circuit 32 to enable the instruction being executed thereby. Thus, in the example of the previous paragraph, addition occurs only during digit 2 since the adder in the arithmetic and register circuit 32 is enabled by the WS line only during this portion of the word. When the WS line is low the contents of the registers in the arithmetic and register circuit 32 are recirculated unchanged. Three examples of WS timing signals are shown in Figure 4. In the first example, digit position 2 is selected out of the entire word. In the second example, the last eleven digits are selected. This corresponds to the mantissa portion of a floating point word format. In the third example, the entire word is selected. Use of the word select feature allows selective addition, transfer, shifting or comparison of portions of the registers within the arithmetic and register circuit 32 with only one basic ADD, TRANSFER, SHIFT or COMPARE instruction. Some customization in the ROM word select fields is available via masking options.

The I_a line serially carries the addresses of the instructions to be read from the ROM circuits Ø–7 of the ROM group 34. These addresses originate from the control and timing circuit 30, which contains an instruction address register that is incremented each word time unless a JUMP SUBROUTINE or a BRANCH instruction is being executed. Each address is transferred to the ROM circuits Ø–7 during bit times b_{19}—b_{26} and is stored in an address register of each ROM. However, only one ROM circuit is active at any given point in time, and only the active ROM circuit responds to an address by outputting an instruction on the I_s line. Control is transferred between the ROMS Ø–7 by a ROM SELECT instruction. This technique allows a single eight-bit address, plus eight special instructions, to address up to eight ROM circuits of two hundred and fifty-six words each.

The START line carries a one-bit pulse which occurs during bit time b_0 and is used to synchronize operations of the data storage assembly 36 and the input-out processor 38 with those of the arithmetic and register circuit 32.

The BCD line carries numerical data between the arithmetic and register circuit 32, the input/output processor 38, and any basic or optional units containing data storage circuits. The format of the data carried on this line is illustrated in Figure 4.

A CARRY line 42 transmits the status of the carry outputs of the adder in the arithmetic and register circuit 32 to the control and timing circuit 30. The control and timing circuit 30 uses this information to make conditional branches, dependent upon the numerical value of the contents of the registers in the arithmetic and register circuit 32.

The control and timing circuit 30 is organized to scan a five-by-eight matrix of switches in search of an interconnection that designates actuation of a key. Any type of metal-to-metal contact may be used as a key. Bounce problems associated with keyboard switch contact are overcome by programmed lockouts in a key entry routine. Each key has an associated six-bit code. To accommodate more than the forty-key maximum represented by the five-by-eight matrix scanner included in the control and timing circuit, an additional keyboard multiplexer circuit is provided. This circuitry is shown in detail in Figure 8.

A standard power supply circuit contains a power-on circuit which supplies a

signal for forcing the calculator to start up in a known condition when power is supplied thereto. A line switch on the calculator keyboard controls the application of operating power. The primary outputs of the calculator are a built-in printer 16 and an optional plug-in LED display unit 14.

CONTROL AND TIMING CIRCUIT

Referring now to Figures 4 and 6, the control and timing circuit 30 contains the master system counter 52, scans the keyboard 12, retains status information about the system or the condition of an algorithm, and generates the next ROM address. It also originates the subclass of word select (WS) signals which involves a pointer 54, a four-bit counter that points to one of the register digit positions.

The control unit of the control and timing circuit 30 is a microprogrammed controller 46 comprising a fifty-eight word (twenty-five bits per word) control ROM, which receives qualifier or status conditions from throughout the calculator and sequentially outputs signals to control the flow of data. Each bit in this control ROM either corresponds to a single control line or is part of a group of N bits encoded into 2^N mutually exclusive control lines and decoded externally to the control ROM. At each phase 2 clock a word is read from the control ROM as determined by its present address. Part of the output is then fed back to become the next address.

Several types of qualifiers are checked. Since most commands are issued only at certain bit times during the word cycle, timing qualifiers are necessary. This means that the control ROM may sit in a wait loop until the appropriate timing qualifier becomes true, and then move to the next address to issue a command. Other qualifiers are the state of the pointer register, the PWO (power on) line, the CARRY flip-flop 76 (Figure 6), and the state of each of the 12 status bits.

Since the calculator is a serial system based on a fifty-six bit word, a six-bit system counter 52 is employed for counting to fifty-six. Several decoders from the system counter 52 are necessary. The SYNC signal is generated during bit times b_{45}—b_{54} and transmitted to the arithmetic and register circuit 32 and all ROM groups present in the calculator system. Other timing qualifiers are sent to the microprogrammed control ROM 56 as mentioned in the previous paragraph.

The system counter 52 is also employed as a keyboard scanner as shown in Figure 7. The three most significant bits of the system counter 52 go to a one-of-eight decoder 58, which sequentially selects one of the keyboard row lines 60. The least significant three bits of the system counter count modulo seven and go to a one-of-five multiplexer 62, which sequentially selects one of the keyboard column lines 64 (during sixteen clock times no key is scanned). The multiplexer output is called the key down signal. If a contact is made at any intersection point in the five-by-eight matrix (by depressing a key), the key down signal will become high for one state of the system counter 52 (i.e., when the appropriate row and column lines are selected). The key down signal will cause that state of the system counter to be saved in a key code buffer 66. This six-bit code is then transferred to a ROM address register 68 and becomes a starting address for the program which services the key that was down (two leading zero bits are added by hardware so an eight-bit address exists). Thus, during each state of the system counter 52, the decoder-multiplexer combination 58 and 62 is looking to see if a specific key is down. If it is, the state of the system counter becomes a starting address for execution of that key function (note that sixteen of the fifty-six states are not used for key codes). By sharing the function of the system counter and using a keyboard scanning technique directly interfaced to the MOS circuitry, circuit complexity is reduced significantly.

While the control and timing circuit 30 is capable of scanning forty keys without the use of additional logic circuitry, the calculator keyboard input unit 12 may be optionally configured with fifty keys. The additional keyboard multiplexing circuitry required to increase the scanning capacity of the control and timing circuit to fifty keys is shown in Figure 8.

A twenty-eight-bit shift register which circulates twice each fifty-six-bit word time is employed in the control and timing circuit of Figure 6. These twenty-eight bits are divided into three functional groups, namely the main ROM address register 68 (eight bits), subroutines return address register 70 (eight bits), and the status register 72 (twelve bits).

The ROM circuits Ø–7 of all the ROM groups present in the calculator as discussed below each contain two hundred fifty-six words of ten bits each, thereby requiring an eight-bit address. This address circulates through a serial adder-

subtractor 74 and is incremented during bit times b_{47}—b_{54} (except in the case of BRANCH and JUMP SUBROUTINE instructions, for which the eight-bit address field of the ten-bit instruction is substituted for the current address). The next address is transmitted over the I_a line to each of the ROM circuits Ø–7 during bit times b_{19}—b_{26}.

The status register 72 contains twelve bits or flags which are used to keep track of the state of the calculator. Such information as whether the decimal point has been hit, the minus sign set, etc. must be retained in the status bits. In each case the calculator remembers past events by setting an appropriate status bit and asking later if it is set. A 'yes' answer to a status interrogation will set the carry flip-flop 76 as indicated by control signal IST in Figure 6. Any status bit can be set, reset, or interrogated while circulating through the adder 74 in response to the appropriate instruction.

The instruction set allows one level of subroutine call. The return address is stored in the eight-bit return address register 70. Execution of a JUMP SUBROUTINE instruction stores the incremented present address into the return address register 70. Execution of the RETURN instruction retrieves this address for transmission over the I_a line. Gating is employed to interrogate the twenty-eight bits circulating in the shift register 68—72 for insertion of addresses at the proper time as indicated by the JSB control signal in Figure 6.

An important feature of the calculator system is the capability to select and operate upon a single digit or a group of digits (such as the exponent field) from the fourteen-digit registers. This feature is implemented through the use of a four-bit pointer 54 which points to the digit of interest. Instructions are available to set, increment, decrement, and interrogate the pointer 54. The pointer is incremented or decremented by the same serial adder/subtractor 74 used for addresses. A 'yes' answer to the IS POINTER N instruction will set the carry flip-flop 76 via the control signal IPT in Figure 6.

The word select feature was discussed above in connection with Figures 3 and 5. Some of the word select signals are generated in the control and timing circuit 30, namely those dependent on the pointer 54, and the remainder in the ROM circuits Ø–7. The pointer word select options are (1) pointer position only and (2) pointer position and all less significant digits. For instance, assume the mantissa signs of the numbers in two registers (i.e. A and C registers) of the arithmetic and register circuit 32 are to be exchanged. The pointer would be set to position 13 (last position) and the A EXCHANGE C instruction with a pointer position word select field would be given. If all of the word except the mantissa signs are to be exchanged, the A EXCHANGE C instruction would be given with the pointer set at 12 and the word select field set to pointer and less significant digits. The control and timing circuit word select (WS) output is OR connected with the ROM word select output and transmitted to the arithmetic and register circuit 32.

Any carry signal out of the adder in the arithmetic and register circuit 32, with word select also high, will set the carry flip-flop 76. This flip-flop is interrogated during the BRANCH instruction to determine if the existing address should be incremented (yes carry) or replaced by the branch address (no carry). The branch address is retained in an eight-bit address buffer 78 and gated to the I_a line by the BRH control signal.

The power-on signal is used to synchronize and preset the starting conditions of the calculator. It has two functions, one of which is to get the address of the control ROM 56 set to a proper starting state, and the other of which is to get the system counter 52 in the control and timing circuit 30 synchronized with the counter in each ROM circuit Ø–7 of the ROM group 34. As the system power comes on, the PWO signal is held at a logical 1 (0 volts in this system) for at least one second. This allows the system counter 52 to make at least one pass through bit times b_{45}—b_{54} when SYNC is high, thereby setting the ROM circuit Ø of the ROM group 34 active and the rest of the ROM circuits inactive. When PWO goes to a logical 0 (+6 volts), the address of the control ROM 56 is set to 000000 where proper operation can begin.

ARITHMETIC AND REGISTER CIRCUIT

The arithmetic and register circuit 32 shown in Figure 9 provides the arithmetic functions and a portion of the data storage for the calculator. It is controlled by the WS, I, and SYNC lines and receives instructions from the ROM circuits Ø–7 of the ROM group 34 over the I_a line, sends information back to the control and timing circuit 30 via the CARRY line 42, partially decodes the display

information before transmitting it via output lines 80 to the anode driver of the output display unit 14, and provides a START pulse to the cathode driver of the output display unit 14 for synchronizing the display, this START pulse used for synchronizing the input/output processor 38 and all data storage circuits employed in the calculator.

The arithmetic and register circuit 32 contains seven fourteen-digit (fifty-six-bit) dynamic registers A—F and M and a serial BCD adder/subtractor 82. Actual data paths, not shown in Figure 4 due to their complexity, are discussed below and shown in Figure 10. The power and flexibility of an instruction set is determined to a great extent by the variety of data paths available. One of the advantages of a serial structure is that additional data paths are not very costly (only one additional gate per path). The structure of the arimethic and register circuit 32 is optimized for the type of algorithms required by the calculator.

The seven registers A—F and M can be divided into three groups: (1) the working registers A, B and C, with C also being the bottom register of a four-register stack; (2) the next three registers D, E, and F in the stack; and (3) a separate storage register M communicating with the other registers through the register C only. In Figure 10, which shows the data paths connecting all the registers A—F and M, each circle represents the fifty-six-bit register designated by the letter in the circle. In the idle state (when no instruction is being executed in the arithmetic and register circuit 32) each register continually circulates since with dynamic MOS registers information is represented by a charge on a parasitic capacitance and must be continually refreshed or lost. This is represented by the loop re-entering each register.

Registers A, B and C can all be interchanged. Either register A or C is connected to one adder input, and either register B or C to the other. The adder output can be directed to either register A or C. Certain instructions can generate a carry via the carry flip-flop 76 which is transmitted to the control and timing circuit 30 to determine conditional branching. Register C always holds a normalized version of the display data.

In the stack formed by the register C, D, E and F a ROLL DOWN instruction is executed by the following transfers: F→E→D→C→D. A STACK UP instruction is executed by the following transfers: G→C→D→E→F. Thus, it is possible to transfer a register and also let it recirculate so that, in the last example, the contents of C are not lost. The structure and operation of a stack such as this are further described in copending U.S. Patent Specification No. 3781820.

In the serial decimal adder/subtractor 82 a correction (addition of 6) to a BCD sum must be made if the sum exceeds nine (a similar correction for subtraction is necessary). It is not known if a correction is needed until the first three bits of the sum have been generated. This is accomplished by adding a four-bit holding register 84 (A_{60}—A_{57}) and inserting the corrected sum into a portion 88 (A_{56}—A_{53}) of the register A if a carry is generated. This holding register 84 is also required for the SHIFT A LEFT instruction. One of the characteristics of a decimal adder is that non-BCD codes, such as 1101, are not allowed. They will be modified if circulated through the adder. The adder logic is minimized to save circuit area. If four-bit codes other than 0000-1001 are processed, they will be modified. This is no constraint for applications involving only numeric data (however, if ASCII codes, for instance, are operated upon, incorrect results will be obtained).

The arithmetic and register circuit 32 receives the instruction during bit times b_{45}—b_{54}. Of the ten types of instructions hereinafter described, the arithmetic and register circuit 32 must respond to only two types, namely ARITHMETIC & REGISTER instructions and DATA ENTRY/DISPLAY instructions. ARITHMETIC & REGISTER instructions are coded by a 10 in the two least significant bits of an I_s register 86. When this combination is detected the most significant five bits are saved in the I_s register 86 and decoded by an instruction decoder 90 into one of thirty-two instructions.

The ARITHMETIC & REGISTER instructions are active or operative only when the word select signal (WS) generated in one of the ROM circuits Ø–7 or in the control and timing circuit 30 is a logical one. For instance, suppose the instruction A+C→C, MANTISSA WITH SIGN ONLY is called. The arithmetic and register circuit 32 decodes only A+C→C. It sets up the registers A and C at the inputs to the adder 82 and, when WS is high, directs the adder output to the register C. Actual addition takes place only during bit times b_{12} to b_{55} (digits 3—13) since for the first three digit times the exponent and the exponent sign are circulating and are directed unchanged back to their original registers. Thus, the word select

signal is an INSTRUCTION ENABLE in the arithmetic and register circuit 32 (when it is a logical 1, instruction execution takes place, and when it is a logical 0, recirculation of all registers continues).

The DATA ENTRY/DISPLAY instructions, except for digit entry, affect an entire register (the word select signal generated in the active ROM is a logical 1 for the entire word cycle). Some of these instructions are: UP STACK, DOWN STACK, MEMORY EXCHANGE M←→C, DISPLAY ON, or DISPLAY TOGGLE. A detailed description of their execution is given hereinafter.

For greater power savings, a display decoder 92 is partitioned to partially decode the BCD data into seven segments and a decimal point in the arithmetic and register circuit 32 by using only five output lines (A—E) 80 with time as the other parameter. Information for seven segments (*a*—*g*) and a decimal point (*dp*) are time shared on the five output lines A—E. The output wave forms for output lines A—E are shown in Figure 11. For example, the output line D carries the segment *e* information during T_1 (the first bit time of each digit time) and the segment *d* information during T_2 (the second bit time of each digit time); the output E carries the segment *g* information during T_1, the segment *f* information during T_2, and the decimal point (*dp*) during T_4. The actual signals which would appear if the digit 9 were decoded are shown in Figure 12. The decoding is completed in the anode driver of the output display unit 14 as explained hereinafter.

The registers in the arithmetic and register circuit 32 hold fourteen digits comprising ten mantissa digits, the mantissa sign, two exponent digits, and the exponent sign. Although the decimal point is not allocated a register position, it is given a full digit position in the output display. This apparent inconsistency is achieved by using both the A and B registers to hold display information. The A register is set up to hold the displayed number with the digits in the proper order. The B register is used as a masking register with the digit 9 inserted for each display position which is to be blanked and the digit 2 inserted at the decimal point location. When the anode driver of the output display unit 14 detects a decimal point code during T_4, it provides a signal to the cathode driver of the output display unit directing a move to the next digit position. One digit and the decimal point share one of the fourteen digit times. The digit 9 mask in the register B allows both trailing and leading zeros to be blanked (i.e., by programming nines into the B register). Use of all three working registers for display (i.e., the C register to retain the number in normalized form, the A register to hold the number in the displayed form, and the B register as a mask) allows the calculator to have both a floating point and a scientific display format at the expense of only a few more ROM states.

The display blanking is handled as follows. At time T_4 the BCD digit is gated from the register A into a display buffer 94. If this digit is to be blanked, the register B will contain a nine (1001) so that at T_4 the end bit (B_{01}) of the B register will be a one (an eight would therefore also work). The input to the display buffer 94 is OR connected with B_{01} and will be set to 1111 if the digit is to be blanked. The decimal point is handled in a similar way. A two (0010) is placed in the register B at the decimal point location. At time T_2 the decimal point buffer flip-flop is set by B_{01}. Any digit with a one in the second position will set the decimal point (i.e., 2, 3, 6 or 7).

The display decoder 92 also applies a START signal to a line 48. This signal is a word synchronization pulse, which resets the digit scanner in the cathode driver of the output display unit 14 to assure that the cathode driver will select digit 1 when the digit 1 information is on the outputs A, B, C, D, and E. The timing for this signal is shown in Figure 13.

One other special decoding feature is required. A minus sign is represented in tens complement notation or sign and magnitude notation by the digit 9 in the sign location. However, the display must show only a minus sign (i.e., segment *g*). The digit 9 in the register A in digit position 2 (exponent sign) or position 13 (mantissa sign) must be displayed as minus. The decoding circuitry uses the pulse on the I_6 line at bit time b_{11} (see Figure 5) to know that the digit 9 in digit position 2 of the register A should be a minus and uses the SYNC pulse to know that the digit 9 in digit position 13 of the register A should also be a minus. The pulse on the I_6 line at bit time b_{11} can be set by a mask option, which allows the minus sign of the exponent to appear in other locations for other uses of the calculator circuits.

READ-ONLY MEMORY CIRCUIT

The ROM group 34, shown in Figure 3, contains eight individual MOS read-only memory circuits, labelled Ø–7 in Figure 16. These circuits store the subroutines required for executing various functions of the calculator system. Each ROM circuit contains 256 ten-bit words, which means a total of 15,360 bits available in the entire ROM group. A detailed diagram of the ROM group 34, showing each of the ROM circuits Ø–7, is given in Figure 16. A block diagram of one of the ROM circuits is shown in Figure 17. Since the ROM circuits Ø–7 are identical except for the way in which they are programmed, only one circuit is shown in the block diagram.

In addition to the ROM group 34 contained within the basic calculator, other ROM groups may be present in a particular calculator configuration by virtue of their presence in various ones of the calculator options. For example, each plug-in keyboard function block contains a ROM group, as does the data storage option and each peripheral input/output interface card.

Figure 44 shows in detail the circuitry required for selecting the ROM group required to perform each given function. The heart of this circuitry is a set of three flip-flops whose outputs are designated A, B, and C. Each of these flip-flops comprises one-half of a dual D-type flip-flop package 184. Several logic gates are employed for controlling the states of these flip-flops. Outputs A, B, and C are connected to the ROM group 34 in the main system, the function block ROM group, and the data storage option ROM group, respectively. These outputs control the clamping gates of their respective ROM groups.

The ROM group associated with each peripheral input/output interface card is responsible for controlling itself and, therefore, includes its own flip-flop. At the time such a ROM group is turned on, the selected peripheral I/O unit will place a negative-true pulse on the line labeled RGC in Figure 44. This causes flip-flops A, B, and C to be reset.

Figure 44 also includes power-on circuitry 186 which provides a signal (PWO) for initializing all of the calculator hardware. PWO will remain low for approximately 1.5 seconds after the power supplies are fully active. This guarantees that the printer motor has come up to speed and that the printer sector counter in the I/O circuit has been synchronized with the printer. In summary, the PWO signal performs the following functions:

1) Inhibits spurious printer operation during start-up
2) Resets printer circuitry contained within the I/O circuit
3) Selects instruction set 1
4) Selects basic calculator system ROM group (ROM group A)
5) Turns on ROM circuit Ø in each ROM group present in the calculator
6) Insures that the first ROM address given after termination of the PWO signal is address zero

Basically, each ROM circuit within the basic calculator ROM group 34 of Figure 16 responds to a serial address input with a serial address output. During every 56-bit word time, an address is inputted, least significant bit first, from bit b_{19} to bit b_{26}. Every ROM circuit Ø–7 of every ROM group present in the calculator receives this same eight-bit address and, from bit time b_{45} to b_{54}, attempts to output onto the I_s line. However, a ROM enable (ROE) flip-flop 96 in each ROM circuit insures that no more than one ROM circuit actually sends an instruction on the I_s line at the same time.

All output signals are inverted so that the steady-state power dissipation is reduced. The calculator circuits are P-channel MOS. Thus, the active signals that turn on a gate are the more negative. This is referred to as negative logic, since the more negative logic level is the logical 1. As mentioned above, a logical 0 is +6 volts and a logical 1 is 0 volts. The signal on the I_s and I_s lines are normally at logical 0. However, when the output buffer circuits are left at logical 0 they consume more power. A decision was therefore made to invert the signals on the I_s and I_s outputs and re-invert the signals at all inputs. Thus, signals appear at the I_s and I_s outputs as positive logic. The oscilloscope pattern that would be seen for instruction 1101 110 011 from state 11 010 101 is shown in Figure 19.

The serial nature of the calculator circuits requires careful synchronization. This synchronization is provided by the SYNC pulse, generated in the control and timing circuit 30 and existing during bit times b_{45}—b_{54}. Each ROM circuit has its own 56-state counter 98, synchronized to the system counter

52 in the control and timing circuit 30. Decoded signals from this state counter 98 open the input to an address register 100 at bit time b_{19}, clock I_a out at bit time b_{45}, and provide other timing control signals.

As the system power comes on, the PWO signal is held at 0 volts (logical 1) for at least one second. The PWO signal is wired (via a masking option) to set the ROM enable (ROE) flip-flop 96 on the ROM circuit Ø of the ROM group 34 and to reset it on all other ROM circuits. Thus, when operation begins, the ROM circuit Ø of the ROM group 34 will be the only active ROM circuit. In addition, the control and timing circuit 30 inhibits the address output during start-up so that the first ROM address will be zero. The first instruction must be a JUMP SUBROUTINE to get the ROM address register 68 in the control and timing circuit 30 loaded properly.

Figure 19 shows the important timing points for a typical addressing sequence. During bit times b_{19}—b_{26}, the address is received serially from the control and timing circuit 30 and loaded into the address register 100 via the I_a line. This address is decoded, and at bit time b_{44} the selected instruction is gated in parallel into an I_s register 102. During bit times b_{45}—b_{54} the instruction is read serially onto the I_s line from the active ROM circuit (i.e., the ROM circuit with the ROM enable flip-flop set).

Control is transferred between ROM circuits by a ROM SELECT instruction. This instruction will turn off the ROE flip-flop 96 on the active ROM circuit and turn on the ROE flip-flop 96 on the selected ROM circuit. Implementation is dependent upon the ROE flip-flop being a master-slave flip-flop. In the active ROM circuit, the ROM SELECT instruction is decoded by a ROM select decoder 104 at bit time b_{44}, and the master portion of the ROE flip-flop 96 is set. The slave portion of the ROE flip-flop 96 is not set until the end of the word time (b_{55}). In the inactive ROM circuits the instruction is read serially into the I_s register 102 during bit times b_{45}—b_{54} and then decoded, and the ROE flip-flop 96 is set at bit time b_{55} in the selected ROM circuit. A masking option on the decoding from the three least significant bits of the I_s register 102 allows each ROM circuit to respond only to its own code.

The six secondary word select signals are generated in the ROM circuits Ø–7 of the ROM group 34. Only the two word select signals dependent upon the POINTER come from the control and timing circuit 30. The word select of the instruction is retained in the word select register 106 (also a master-slave). If the first two bits are 01, the instruction is of the arithmetic type for which the ROM circuit must generate a word select gating signal. At bit time b_{55} the next three bits are gated to the slave and retained for the next word time to be decoded into one of six signals. The synchronization counter 98 provides timing information to the word select decoder 108. The output WS signal is gated by the ROE flip-flop 96 so that only the active ROM circuit can output on the WS line, which is OR connected with all other ROM circuits and also with the control and timing circuit 30. As discussed above, the WS signal goes to the arithmetic and register circuit 32 to control the portion of a word time during which an instruction is active.

The six ROM-generated word select signals used in the calculator are shown in Figure 20. The ROM circuits Ø–7 output a single pulse on the I_s line at bit time b_{11} to denote the exponent minus sign time. This pulse is used in the display decoder of the arithmetic and register circuit 32 to convert a 9 into a displayed minus sign. The time location of this pulse is a mask option on the ROM circuit.

INPUT/OUTPUT CIRCUIT

The input/output (I/O) processor 38 of Figure 3 is shown in more detail in Figure 21. It includes an MOS/LSI input/output circuit 110, a one-of-sixteen decoder 112, and various other logic gates. The I/O processor serves as a binary arithmetic logic unit, drives the printer unit (Seiko model 102), and controls the flow of instructions and addresses between the arithmetic and register circuit, the control and timing circuit, and the read-only memory group.

The I/O processor 38 is linked to the arithmetic and register circuit and to the control and timing circuit by eight signals, which are (1) input clock phase ($Ø_1$), (2) output clock phase ($Ø_2$), (3) power on (PWO), (4) instruction line (I_s), (5) timing (START), (6) data bus (BCD), (7), flag line (FLG), and (8) alternate ROM address input (EXT).

The I/O circuit 110 includes twenty outputs (C1—C20) comprising parallel data signals used by the printer unit. C13 controls the printer solenoid for column 13. C19 and C20 control the PRINT, STANDBY, ADVANCE PAPER, and RED

commands. Outputs C9—C18 are employed as memory address lines when a ROM, PROM, or read/write memory is plugged into the calculator. Output lines C1—C8 are bidirectional and are used as inputs when the LATCH line is grounded. Output lines C1—C18, along with the LATCH and GIOE lines, are available at each I/O peripheral slot to be used as needed by each peripheral unit, as described by the interface software associated with each such peripheral unit.

The EISI line of Figure 21 is used as a gate control for the instruction line I_8. It also comprises one of the inputs, along with lines IOC1, IOC2, and IOC3, to the one-of-sixteen decoder 112. Switching transients might be expected to appear as the various IOC lines change state. To eliminate this problem, a decoder strobe signal SCE is issued by the I/O circuit 110. By means of this technique, a total of fourteen different command signals are received from the decoder 112 as pulses of approximately 200 microsecond duration. These signals are described in the table below.

Decoder Pin #	Function
1	Interrogate PRINT OFF key
2	Interrogate function block keyboard flag
3	Turn on busy light
4	READ command for plug-in ROM or PROM
5	Interrogate AUTO DECIMAL key
6	Select ROM group for data storage option
7	Select ROM group for function block
8	Return control to basic ROM group
9	TG8 (reserved for control of peripherals)
10	TG9 (reserved for control of peripherals)
14	General I/O device enable
15	Turn off busy light
16	WRITE command for read-write memory
17	YINTF instruction reserved for control of peripherals

A light emitting diode (LED) 114 shown in Figure 21 is used as a busy light. When lit, it indicates that the calculator is busy and cannot accept keyboard commands. The busy light will be turned on at the onset of the power-on sequence, during execution of one of the routines involving exponentiation, or during execution of a library program. Upon completion of any of these functions the busy light will be turned off, and the calculator will then accept keyboard entries.

The status of the PRINT OFF and AUTO DECIMAL keys as well as the FPS line, which originates in the keyboard multiplexer, is examined by the decoder 112. If the appropriate signal is present the FLG' line will be pulled low for approximately 200 microseconds. The FLG' line may be used by the interface circuitry associated with various input/output peripheral units.

Input lines TP and TR to the I/O circuit 110 are timing signals received from the printer.

The MOS input/output circuit 110 of Figure 21 is shown in detail in the block diagram of Figures 22A—B. An instruction set switching scheme has been implemented in the I/O circuit to achieve greater efficiency.

The instruction lines of the ROM circuits Ø–7 of the ROM group 34 are tied directly to the I_s input to the I/O circuit. The I/O circuit issues a signal EISl. When this signal is high, the instruction lines of the ROM circuits are fed to the I_s line going to the control and timing circuit 30 and to the arithmetic and register circuit 32. When EISl is low, then I_s line going to the control and timing circuit and to the arithmetic and register circuit is held low, and the I/O circuit is enabled to give instructions with the same bit patterns as used normally by the control and timing circuit and the arithmetic and register circuit. Since the I/O circuit has the ability to remember in which instruction set mode it is operating, it doesn't try to execute those instructions designated as IS2 instructions when the calculator is in the ISl mode. The power-on (PWO) line to the I/O circuit insures that the I/O circuit powers up in the ISl instruction set. When ISl is executed, an internal IS2 instruction flip-flop 114 is reset, which implies that EISl will go high and the I/O circuit will accept only those instructions whose bit patterns are the same in all instruction sets. When IS2 is executed, EISl will go low, and the IS2 instruction flip-flop 114 will be set.

The input/output circuit includes an instruction register 116 which converts the serial instruction coming from the ROM group 34 on the I_s line to a parallel instruction for decoding by an instruction decoding ROM 118. A nine-bit shift register 120 and a one-bit shift register 122 serve as an output register for the I/O circuit.

The I/O circuit 110 also includes a 56-bit shift register 124, called the T-register. This register serves as the primary working register when the I/O circuit is called upon to perform binary arithmetic operations. Logic gates 126 and 128 form a bidirectional buffer for the BCD line, allowing binary data to be inputted to or outputted from the I/O circuit by means of the BCD line. A binary processor ROM 130 serves as an arithmetic logic unit for performing all the basic binary operations on the contents of the T-register. It also helps control data transfers into the T-register. In performing binary arithmetic with the I/O circuit, a carry on line 132 may result from a particular operation. This condition causes a signal on the FLG line and sets status bit eleven in the control and timing circuit 30.

The I/O circuit 110 further comprises a 14-bit register 133, known as the P-register. This register is available to the user as a 14-bit storage register, but is generally employed by the calculator system as a program counter. The fourteen least significant bits in the T-register 124 can be loaded into the P-register 133, without altering the contents of the T-register 124. The contents of the P-register 133 can be loaded into the fourteen least significant bits of the T-register 124 by means of the PTT instruction. In this case, the forty-two most significant bits of the T-register and the entire contents of the P-register are unaltered.

The PINC instruction performs a binary increment on the number stored in the P-register. Not only is the incremented binary number placed in the P-register 133, but also the ten least significant bits of the incremented number are loaded sequentially onto the output lines C9—C18. The PDEC instruction is similar to PINC except that the binary number stored in the P-register is decremented by one. The instructions PINC and PDEC have a common restriction in that if the decimal equivalent of the number in the P-register exceeds 1,023, the number displayed on the output lines C9—C18 will not only have an insufficient number of bits, but the bit displayed on the line C18 will always be a logical 1. An arithmetic logic unit 134 is employed to perform the increment and decrement operations called for by PINC and PDEC.

The parallel data output lines Cl—C18 are low power TTL compatible. The lines C1—C8 may be used to either input or output data, whereas the lines C9—C18 are outputs only. When the LATCH input line to the I/O circuit is high, the line C1—C8 become outputs, and when the LATCH is low, these lines become inputs. The data found on the lines C1—C8 when the LATCH is held low is transferred into an output buffer register 136. After the LATCH is released the data loaded will remain as an output on the lines C1—C8.

The data on the lines C1—C8 may be operated on by two instructions, EERA and IXT. These instruction lines are shown as 138 and 140, respectively, in Figures 22A—B. The instruction EERA is understood by both the control and timing circuit 30 and by the I/O circuit 110. When this instruction is executed, the eight-bit word on the lines C1—C8 is sent serially on the EXT output line and during the same period of time as the addresses are sent to the ROM group 34. The control and timing circuit 30 recognizes the EERA command and transfers the data from its EXT input line to its 1A output line. In this way the next ROM address is

obtained from the I/O circuit. After executing EERA, bits C1—C8 will contain the same data (either zeros or ones) as bit C9.

The instruction IXT exchanges the data on the lines C1—C8 with the eight most significant bits of the T-register 124. The remaining bits of the T-register 124, as well as the lines C9—C18 of the I/O register 120, are unaltered.

The I/O circuit 110 also contains the necessary logic circuitry and memory for driving the printer unit 16, which is a Seiko model 102. This printer has eighteen columns and a choice of thirteen characters per column. It includes a rotating print drum having thirteen sectors. Two timing signals are required from the printer as inputs to the I/O circuit 110. They are labelled on Figures 22A—B as TP and TR and are clocked with phase two of the clocks.

During operation the print drum is continually rotating, and the current sector of the printer is continually noted by a sector counter 142. During the portion of time that the printer is enabled the end of each sector is marked by a flag pulse which sets status bit eleven in the control and timing circuit 30. The method of obtaining a print begins with the construction of a printer mask in a C-register 144 of the arithmetic and register circuit 32 and the T-register 124 of the I/O circuit 110. The columns 1—14 of the printer (right to left) are printed according to the data in the C-register. Each digit position is filled with a hexadecimal number (0—15) which represents the sector to be printed. A hexadecimal 13, 14 or 15 will leave a blank in the given column position. Printing is best accomplished when the numeric entries on the print drum coincide with their respective sector numbers. The four least significant digits of the T-register 124 are loaded in a similar manner for columns 15—18.

The RED instruction preceding the print subroutine shifts the ribbon into the red printing position. The ribbon will remain in that position until a paper advance occurs. A paper advance instruction (ADV) is used to terminate each print sequence. The output format for driving the printer is such that during the period of time that the printer is enabled a logical zero on the lines C1—C18 indicates that the column solenoid under consideration should be energized. The outputs on the lines C19 and C20 control the modes of operation of the printer as shown in the table below.

C19	C20	Mode
Low	Low	Printer Standby
Low	High	Energize Ribbon Solenoid
High	Low	Energize Paper Solenoid
High	High	Enable Printing Solenoids

The portion of the I/O circuit 110 which relates to the printer also includes a four-bit shift register 148 which receives data to be printed from either the T-register 124 of the I/O circuit 110 or the C-register 144 of the arithmetic and register circuit 32 and presents it in parallel form to a comparator 150. Data select logic circuitry 146 transmits data from either the C-register or the T-register to the shift register 148. The comparator 150 functions to compare the current sector count received from the sector counter 142 with the output of the shift register 148 to determine when the print drum is in the proper position to print each particular data item. A printer controller 152 receives printer-related instructions from the instruction decoding ROM 118 and, in response thereto, controls the print sequence.

The I/O circuit 110 further comprises a master clock 154 which supplies the required timing signals within the I/O circuit. Also included is a Schmitt trigger 156 which receives FLG' signals from peripheral input/output units employed with the calculator and properly gates these with clock signal Ø2 from the control and timing circuit 30.

DATA STORAGE CIRCUIT

The data storage assembly 36 of Figure 3 is shown in more detail in Figure 24. It includes an MOS/LSI data storage circuit 158 and a bidirectional amplifier 160.

The data storage circuit 158 contains ten data storage registers. In addition to the single data storage circuit employed in the calculator as basically configured, the user may optionally configure the calculator to include additional data storage. Also, data storage circuits are included as integral components within the optional plug-in keyboard function blocks and the optional peripheral input/output interface cards associated with various peripheral I/O units which might be employed with the calculator.

The BCD lines associated with the arithmetic and register circuit 32 and with the I/O circuit 110 are directly connected to each other and could, in turn, be directly connected to as many as three additional data storage circuits. The bidirectional amplifier 160 is provided to drive the excessive capacitance seen by the BCD line when more than three data storage circuits are employed. Since the BCD line carries binary data in both directions, the amplifier 160 must have bidirectional capability. In order to control the direction of data flow through the amplifier, a steering signal (BDE) is required. Normally, this signal will be low (logical 0), in which case the BCD line from the arithmetic and register circuit 32 and from the I/O circuit 110 is the input, while the BCD line connected to the data storage circuits is the output.

A DATA STORAGE TO C-REGISTER instruction (DSTC) is decoded by all the data storage circuits present in the current configuration of the calculator. This would include the data storage circuit in the basic calculator as well as those currently employed in a keyboard function block, optional data storage, and peripheral input/output interface cards. Even through only the previously addressed data storage circuit executes the DSTC instruction, all of the data storage circuits present will respond by bringing the BDE line to a logical 1 level during the entire word in which the data transfer occurs. By using this technique only the BDE line from the single data storage circuit in the basic calculator is required to control the direction of the amplifier 160 for all of the data storage circuits.

The MOS data storage circuit 158 of Figure 24 is shown in detail in the block diagram of Figures 25A—B. The data storage circuit 158 recognizes three separate instructions which are received directly from the active ROM circuits via the I_2 line. These are (1) ADDRESS FROM C-REGISTER TO DATA STORAGE (ATDS), (2) DATA FROM C-REGISTER TO DATA STORAGE (DTDS), and (3) READ FROM DATA STORAGE INTO C-REGISTER (DSTC). These serial instructions are received by a ten-bit shift register 162, converted into a ten-bit parallel signal, and transmitted to an instruction decoder 164.

The contents of the C-register in the arithmetic and register circuit 32 are continuously displayed on the BCD line, except when one of the above instructions is issued. When a data transfer instruction occurs, that instruction will only be executed if there exists at least one data storage register which has previously been enabled. Once enabled, a register will remain enabled until another data storage transfer instruction is issued which addresses a different register.

The designation of each register comprises two separate parts. The ten registers within each data storage circuit are numbered 0, 1, 2, . . . 8, 9. The data storage circuits within a particular calculator configuration are themselves numbered 0, 1, 2, 3, . . . 62, 63. Each data storage circuit may be coded in any combination, thereby eliminating the necessity of different data storage circuits for association with different addresses. The circuit address bits, in ascending order of significance, are designated as B1, B2, B3, B4, A1, A2. If the bits B1—B4 are considered as a hexadecimal number, then only ten of the sixteen possible combinations are allowable BCD characters. Normal addresses are those utilizing the ten allowable BCD codes for bits B1—B4. The special addresses are the remaining combinations. Normal register addresses are designated by $0 \leq D_3D_2D_1 \leq 399$, where all three digits are allowable BCD numbers. If D_2 is restricted to the hexadecimal numbers ten to sixteen, then 240 addresses are available.

To address a data storage circuit it is necessary to entry the binary (or BCD) code of the appropriate circuit number into the A and B inputs to a six-bit comparator 166. For example, to establish the register addresses for a circuit as 130 to 139 would mean that the circuit should be identified as circuit 13. The BCD code for 13 is 01 0011. Therefore, the inputs to the comparator 166 would be as shown below.

A2 = 0

A1 = 1

B4 = 0

B3 = 0

B2 = 1

B1 = 1

The address decoding logic of the data storage circuits is capable of handling floating point addresses. In order that an address be accepted by a data storage circuit the exponent must be either +0, +1 or +2. The sign of the mantissa is always ignored by a data storage circuit. In addition, the addresses are truncated to the next lower integer. For example, the numbers –3.79, 3.1854, +3.00, –3.00 would, as addresses, enable register three. If an exponent equal to zero is given, only the most significant digit of the mantissa will be interrogated to determine a register number in the range 0—9. If an exponent equal to +1 is found, the two most significant digits of the mantissa are interrogated to determine a register number in the range 0—99. When an exponent equal to +2 is encountered in an address, the three most significant digits of the mantissa are interrogated and checked for a number in the range 0—399.

The I/O circuit also includes four sample and hold flip-flops 168 for storing the desired register (0—9) within a particular data storage circuit. A one-of-ten decoder 170 actually selects the desired register within the circuit. An exponent = 0 line 172, an exponent = 1 line 174, and an exponent = 2 line 176 serve to indicate whether the exponent of a floating point number used as an address was decoded as a 0, 1, or 2, respectively. For purposes of illustration only two of the ten 56-bit storage registers comprising each data storage circuit have been shown in Figures 25A—B as items 178 and 180. All ten of the registers are identical in construction. Also included within the data storage circuit 158 is a sample and hold flip-flop 182 for remembering whether or not a register previously addressed is locataed within a given data storage circuit.

Several signals associated with the data storage circuit 158 and their relative waveforms with respect to a 56-bit word time are shown in Figure 26.

CLOCK DRIVER

A bipolar clock driver 188, one phase of which is shown in Figure 14, requires less than 25 milliwatts of power and can drive loads up to three hundred picoforads with a voltage swing of +7 to –14 volts. An ENABLE input 190 allows both outputs Q_1 and Q_2 to be held to V_{CC}, the MOS logical 0. This is an effective means of strobing the clock. During DC operation, the transistor pair Q_1—Q_2 allows only one of the output transistor pairs Q_5—Q_6 or Q_7—Q_8 to conduct. Diode D_3 prohibits conduction from the transistor Q_6 to the transistor Q_8 during transient operation. Thus, the only possible transient short circuit current must flow from the transistor Q_5 to the transistor Q_7. However, the limited current handling capability of the transistor Q_5 (a lateral PNP) limits this current to less than five milliamps peak. The input signals for the clock driver 188 are generated on the anode driver of the optional output display unit 14, or, if the display unit is not employed in the calculator, then by means of the clock generator circuitry shown in Figure 40. The outputs of the clock driver are connected to each of the MOS/LSI circuits in the calculator. The timing relationship of the input and output signals of the clock driver circuit are shown in Figure 15.

LED DISPLAY

An optional plug-in light emitting diode (LED) output display unit 14 may be employed with the calculator. As shown in the block diagram of Figure 29, the display unit comprises an anode driver 192, a cathode driver 194, and three 5-digit LED clusters 196.

The display unit employs an inductive drive technique which is inherently efficient because of the absence of components representing large power losses.

The only dissipation is created by the parasitic resistances and the saturated transistor switches. An inductive driver like that used in the calculator is shown and described in copending U.S. Patent Specification No. 3755697.

The display circuitry used in the calculator is shown in Figure 36. It comprises an 8 x 15 array of LEDs in which the eight rows are scanned by the anode driver and the fifteen columns by the cathode driver. A simplified circuit diagram for a segment of the display involving one LED is shown in Figure 37. The equivalent piecewise-linear circuit model is shown in Figure 38. An analysis of this model shows the inductor current buildup and discharge to be nearly linear for the parameters used in the calculator. The discharge-time to charge-time ratio is approximately:

$$\frac{t_{discharge}}{t_{charge}} \quad \frac{V_s - V_{asat}}{V_s + V_{csat}} = \frac{3.8 - 0.1}{1.6 + 0.2} = \frac{3.7}{1.8} = 2.06$$

Figure 38 shows the inductor current for a basic calculator clock frequency of 175 KHz. The average LED current can be calculated from the formula

$$\text{Ave } I_{LED} = \text{pulse current} \times \text{duty cycle}$$

$$= \left(\frac{1}{2} \times 80 \text{ ma}\right) \frac{5.88 \text{ sec}}{\frac{1}{175 \text{ KHz}} \times 56}$$

$$= \frac{(80)\ (5.88)\ (.175)}{(2)\ (56)} = .735 \text{ ma}$$

The worst case display power (i.e., thirteen figure eights and two minus signs) is about 110 milliwatts. Figure 38 also shows the ringing inherent in the inductor drive technique.

The display information is partially decoded in the arithmetic and register circuit 32 and is completely decoded into eight signals, representing seven character segments plus a decimal point, within the bipolar anode driver 192 of the output display unit 14.

As discussed above, the anode driver 192 includes the basic clock generator for the calculator. In the event the optional display unit is not employed, the clock signals are derived from separate clock generator circuitry located elsewhere in the basic calculator. This separate clock generator circuitry is shown in detail in Figure 40. Low power TTL inverters 198 and the associated passive elements form an oscillator whose frequency is, under worst case conditions, always less than or equal to 720 kilohertz. The oscillator output is a square wave. A dual J—K master-slave flip-flop 200 operates on the output of the oscillator to provide output signals comprising 360 kilohertz and 180 kilohertz square waves. The resulting clock phase signals Ø1T and Ø2T appearing at the outputs of gates 202 are identical to the signals generated by the clock generator portion of the anode driver 192.

A logic diagram of the anode driver 192 is shown in Figure 30. The clock generator portion thereof uses an external LC circuit to set the oscillator frequency at a nominal 800 kilohertz rate. Flip-flops B1 and B2 are clocked off alternate phases of the flip-flop B1 to provide two 200 kilohertz square waves as shown in Figure 31. Flip-flop B3 is clocked from the flip-flop B2 and in turn clocks a flip-flop B4 to provide further division of the basic clock frequency of approximately 200 kilohertz. The two-phase clock signals Ø1T and Ø2T are generated from the flip-flops BL and B1 and the 800 kilohertz oscillator 200. These signals are on for 625 nanoseconds and are separated by 625 nanoseconds, as shown in Figure 31. A periodic counter-clock signal (CCL) is also derived from the anode driver 192 and is sent to the cathode driver 194 once each digit time. The trailing edge of this signal causes the display to step to the next digit.

The display consists of fifteen characters while the basic calculator word cycle

consists of fourteen digits. The extra character is the decimal point. As explained above, a BCD two is placed in the register B at the digit position of the decimal point. The display decoder 92 in the arithmetic and register circuit 32 indicates this by a signal on outputs B and E during bit time T_4 (see Figure 11). When this condition is decoded by the anode driver, the decimal point is excited and an extra counter clock signal is given to step the display to the next position (see Figures 31, 32, and 33). Therefore, all remaining digits in register A are displaced one digit in the display.

Figures 32 and 33 show the simplified circuit and the timing relationship for the decimal point. The timing is critical since all the inductor current in segment *b* (the last to be excited) must be decayed before the counter clock signal is given to step to the next digit, or the remaining current would be discharged through the wrong digit and a faint lighting of the segment *b* on the same digit with the decimal point would occur. The decimal point insertion technique is the reason all other seven segments are excited during the first half of the digit time. The decimal point charging time is one-half that of the other segments. The decimal point segment receives the same current in one-half the time and is one-half as bright as the other segments.

As described above, an inductive circuit method of driving the light emitting diodes is employed. Basically, the method involves using the time required for current to build up in an inductor to limit current, rather than using a resistor as is normally the case with LED displays. This saves power since the only lossy components in the drive system are the parasitic inductor and transistor resistances. The drive circuit for one digit is shown in Figure 34. Assuming the cathode transistor switch T_c is closed, an anode switch T_a is closed for 2.5 microseconds, allowing the current to build up to a value I_p along a nearly triangular waveform (the early part of an exponential buildup). When the anode switch T_a is opened, the current is dumped through the LED, decaying in about 5 microseconds. The anodes are strobed according to the sequence in Figure 31. The primary reason for sequentially exciting the anodes is to reduce the peak cathode transistor current. Since the decay time is approximately twice the buildup time, it works out that the peak cathode current is about 2.5 times the peak current in any segment. The LEDs are more efficient when excited at a low duty cycle. This means high currents for short periods (80 ma. anode current, 250 ma. cathode current). Figure 31 also shows the relationship between the anode strobing sequence and the display output signals (A—E) from the arithmetic and register circuit 32.

The cathode driver 194 of the output display unit 14 comprises a fifteen-position shift register for scanning the fifteen-digit display once each word time. This scanning operation moves from digit to digit in response to counter clock signals from the anode driver. Once each word time a START signal arrives from the arithmetic and register circuit 32 to restart the procedure. A block diagram of the cathode driver 194 is shown in Figure 35.

OUTPUT PRINTER

As discussed above, primary control for the output printer unit 16 is provided within the MOS/LSI input/output circuit 110. However, some additional circuitry outside the I/O circuit is provided to perform timing and driving functions relative to the output printer.

Figure 27 is a detailed schematic diagram of the printer timing circuitry. The I/O circuit 110 requires a pulse train (TP) from the printer which denotes the sector divisions as the print drum rotates. Between sectors twelve and zero of the print drum, a RESET signal is given by the printer to the printer timing circuitry, which responds by issuing a signal (TR) to the I/O circuit for resetting the sector counter 142.

Since both RESET and TP are derived from a magnetic pick-up within the printer, they have very slow rise and fall times. In addition, they are very low level signals. The printer timing circuitry of Figure 27 operates on these signals to present them in better form to the I/O circuit. Operational amplifiers 206 serve as level detectors. In order to mask the problem of spurious pulses resulting from the fact that signal TP has very little magnitude, a retriggerable monostable multivibrator 208 is provided to receive the output signal of the operational amplifier 206. Since both signals TP and TR must be received by the I/O circuit 110 in synchronization with phase 2 of the clock signal, a dual D-type flip-flop 210 is provided.

The printer drive circuitry of Figure 28 simply accepts the low level TTL signals of the I/O circuit 110 for interfacing to the various solenoids of the printer 16, which require higher power levels for operation.

POWER SUPPLY

The calculator power supply system, shown in detail in Figure 41, is constructed according to conventional practices and delivers +6 volts, +15 volts, +5 volts, +7.5 volts, and −12 volts to the various calculator circuits.

PLUG-IN KEYBOARD FUNCTION BLOCK

Figure 45 shows a detailed schematic diagram of the circuitry comprising each plug-in keyboard function block. Each such block contains two of the MOS/LSI data storage circuits 158 which are constructed and accessed by the calculator microprograms as described above. Also included are eight MOS/LSI ROM circuits Ø–7 which form a ROM group 212. Again, these circuits are identical to those of the basic calculator ROM group 34 except for their bit patterns, which are set forth in the instruction listing at a later point in this specification. The function block circuitry also includes a bipolar clock driver 188 which is identical to that described above.

PLUG-IN ROM/PROM OPTION

As described above, the user may employ with the calculator a plug-in ROM or PROM containing library programs written in user level language. These programmes may be executed by the user from the keyboard. A detailed schematic diagram of each plug-in ROM/PROM is given in Figure 46. Included are four conventional 256-bit read-only memories or programmable read-only memories 214, in which are stored the instructions comprising the library program. The advantage of using a programmable read-only memory (PROM) lies in the fact that the program instructions may be modified. A decoder 216 selects which of the ROMs or PROMS 214 is to be addressed, based on the two most significant bits of the total address. A transistor switch 218, in response to a READ instruction, pulls the LATCH input of the I/O circuit 110 low, thereby allowing data from the ROMs or PROMs 214 to be loaded into the I/O circuit. Logic circuitry 222 issues a flag to the I/O circuit 110 whenever a given address exceeds the memory capacity of the ROM/PROM option currently employed. A transistor switch 234 provides an output, delayed in time from that of the transistor 218, for enabling the decoder 216. Regulator circuitry 226 provides additional regulation of two of the power supply voltages as they are applied to the ROM/PROM circuitry. A logic gate 228 is employed by some plug-in keyboard function blocks to determine if the program stored in a ROM or PROM 214 is proprietary, in which case program listing is prohibited.

DATA STORAGE OPTION

The optional plug-in data storage, described above, is shown in detailed schematic form in Figure 47. Each increment of plug-in data storage comprises thirty registers, which are provided by three MOS/LSI data storage circuits 158. These circuits are identical in construction and function to that described in detail earlier in this specification. Three MOS/LSI ROM circuits, like those described in detail earlier in this specification, comprise a ROM group 230. The bit patterns for these individual ROM circuits are contained in the instruction listing below. The data storage option also includes a clock driver 232 as described above.

INSTRUCTION SET 1

Every function performed by the calculator is implemented by a sequence of one or more ten-bit instructions stored in the ROM circuits Ø–7 of the ROM group 34 or one of the other ROM groups located elsewhere in the calculator such as a keyboard function block 20, an optional data storage unit, or one of the optional input/output interface cards. The serial nature of the MOS calculator circuits allows the instruction bits to be decoded from least significant bit to most significant bit (right to left) serially. If the first bit is a one, the instruction is either a subroutine jump or a conditional branch as selected by the second bit, with eight bits remaining for an address. The next largest set of instructions, the arithmetic set, starts with a zero followed by a one (right to left), leaving eight bits for encoded instructions.

With three exceptions, instruction set 1 includes those instructions executed

by the arithmetic and register circuit, the control and timing circuit, and the read-only memory circuits. These three exceptions are represented by three instructions within instruction set 1 which are executed by the data storage circuit. The mnemonics and associated bit patterns for these three instructions are as follows:

MNEMONIC	BIT PATTERN
DSTC	1 0 1 1 1 1 1 0 0 0
ATDS	1 0 0 1 1 1 0 0 0 0
DTDS	1 0 1 1 1 1 0 0 0 0

These instructions are discussed in detail in the portion of this specification above which deals with the MOS/LSI data storage circuit.

The ten different types of instructions comprising instruction set 1 are shown in the following table.

TABLE OF INSTRUCTION TYPES (X = DON'T CARE)

TYPE	AVAILABLE INSTRUCTIONS	NAME	FIELDS
1	256 (ADDRESSES)	JUMP SUBROUTINE	8 bits: SUBROUTINE ADDRESS | 0 1
	256 (ADDRESSES)	CONDITIONAL BRANCH	BRANCH ADDRESS | 1 1 (I_9 ... I_0)
2	32 × 8 × 256	ARITHMETIC/REGISTER	5: OPERATION CODE | 3: WORD SELECT | 1 0
3	64 (37 used)	STATUS OPERATIONS	4: N | 2: F | 0 1 0 0 (I_5I_4 I_3 I_2 I_1 I_0) SET BIT N: F = 00 INTERROGATE N: F = 01 RESET N: F = 10 CLEAR ALL: F = 11 (N = 0000)
4	64 (30 used)	POINTER OPERATIONS	4: P | 2: F | 1 1 0 0 SET POINTER TO P: F = 00 INTERROGATE P: F = 10 DECREMENT P: F = 01 INCREMENT P: F = 11 P = XXXX
5	64 (20 used)	DATA ENTRY/DISPLAY	4: N | 2: F | 1 0 0 0 LOAD CONSTANT: F = 01 $I_S \rightarrow A$: F = 1X (N = XX01) BCD INPUT TO C-REGISTER: F = 1X (N = XX11) STACK INSTRUCTIONS: F = 10 N = (---0) AVAILABLE: F = 00
6	32 (11 used)	ROM SELECT, MISC.	3: N | 2: F | 1 0 0 0 0 SELECT ROM N: F = 00 KEYBOARD ENTRY: F = 10 (N = XX1) EXTERNAL ENTRY: (N = XX0) SUBROUTINE RETURN: F = 01 (N = XXX)
7	16	(RESERVED FOR PROGRAM STORAGE	4: X X X X | 1 0 0 0 0 0
8	8	MOS CIRCUIT)	3: X X X | 1 0 0 0 0 0 0
9	7	AVAILABLE	X X X | 0 0 0 0 0 0 0
10	1	NO OPERATION (NOP)	0 0 0 0 0 0 0 0 0 0

There are two type 1 instructions, JUMP SUBROUTINE and CONDITIONAL BRANCH. They are decoded only by the control and timing circuit 30. No word select is generated and all registers in the arithmetic and register circuit 32 merely recirculate. The object of the JUMP SUBROUTINE instruction is to move to a new address in the ROM circuit which is currently active and to save the existing address, incremented by one, as a return address. The last instruction in a subroutine must be a RETURN to continue the program where it was left previously.

As discussed above, the control and timing circuit 30 contains a 28-bit shift register 68—72 which holds the current 8-bit ROM address and also has eight bits of storage for one return address (see Figure 6). During the bit times b_{47}—b_{54} the current ROM address flows through the adder 74 and is incremented by one. Normally this address is updated each word time. However, if the first two bits of the instruction, which arrive at the bit times b_{45}—b_{46}, are 10, the incremented current address is routed to the return address register 70 of the 28-bit shift register, and the remaining eight bits of the instruction, which are the subroutine address, are inserted into the address register 68. These data paths with the JSB control line are shown in Figure 6. In this way the return address has been saved and the jump address is ready to be transmitted to the currently active ROM circuit at bit times b_{19}—b_{26} of the next word.

The most frequently used instruction is the CONDITIONAL BRANCH, which, based upon data or system status, implements the decision-making capability of the calculator. In the calculator system described here this instruction also functions as an unconditional branch.

The format of the CONDITIONAL BRANCH instruction, as shown in the instruction table above, is two ones followed by an 8-bit branch address. The instruction is received at bit times b_{45}—b_{54}. The last eight bits of the instruction are stored in the address buffer 78 (See Fig. 6). During the next word time the carry flip-flop 76 is checked at bit time b_{19}. If the carry flip-flop was set during the previous word time, the current ROM address is transmitted to the ROM circuits Ø–7. If the carry flip-flop was not set, the branch address is read from the address buffer 78 onto the I_a line and loaded into the ROM address register 100 (see Fig. 17). Thus, the instruction causes a branch if there is no carry. There are three ways the carry flip-flop 76 can be set: (1) by a carry generated in the arithmetic and register circuit 32; (2) by a successful interrogation of the pointer position; and (3) by a successful interrogation of one of the twelve status bits. An example is given in the table below.

EXAMPLE OF CONDITIONAL BRANCH EXECUTION

WORD	ADDRESS RECEIVED AT ROM	INSTRUCTION SENT BY ROM	INSTRUCTION EXECUTED	RESULT
N–1	P	INCREMENT SIGN DIGIT	—	—
N	P+1	CONDITIONAL BRANCH TO ADDRESS Q	INCREMENT SIGN DIGIT	CARRY GENERATED IF A–REGISTER NEGATIVE
N+1	P+2	CONTENTS OF P+2	CONDITIONAL BRANCH	SEND P+2
	or	or		or
	Q	CONTENTS OF Q		SEND Q

A typical test condition is to determine the sign of a number. Suppose at address P in the program a branch to location Q is desired if the sign of A is positive, while program execution is to continue if the sign is negative. In the

example given in the table above, the instruction INCREMENT THE A-REGISTER, WORD SELECT OF SIGN DIGIT ONLY is given at location P. During word time N—1 the instruction is received by the arithmetic and register circuit 32 and is executed at word time N (the same word time when the CONDITIONAL BRANCH instruction is received by the control and timing circuit 30). If the sign of A is negative, there will be a nine in the sign digit. Incrementing this position will generate a carry and set the carry flip-flop 76 in the control and timing circuit 30. Since the instruction causes a branch if no carry is generated, the program execution will jump to location Q only if the sign is positive (i.e., was a zero), otherwise execution continues at P+2.

Note that during word time N+1 the calculator did nothing more than select which of two addresses to send next (all registers merely recirculate). Performing a branch actually takes two word cycles to execute; one to ask a question and set the carry flip-flop 76 if the answer is YES, and the other to test if the carry flip-flop was set and transmit the proper address. In many cases, asking the question is an arithmetic operation (i.e., A+B→A) which must be performed anyway. In this case, the branch requires only one extra instruction.

Contrary to most instruction sets, this set has no unconditional branch instruction. However, since an ordinary JUMP is one of the most frequently used instructions, the CONDITIONAL BRANCH is also used to effect an unconditional branch or jump by insuring that the carry flip-flop 76 is reset when an unconditional branch is desired. This is the reason the sense of the CONDITIONAL BRANCH is branch on no carry. The carry flip-flop 76 is reset during execution of every instruction except ARITHMETIC (type 2) and INTERROGATION OF POINTER or STATUS (types 3 and 4). Since only ARITHMETIC and INTERROGATION instructions can set the carry flip-flop 76, the constraint is not severe. The JUMP SUBROUTINE instruction can also be used as an unconditional branch if the previous return address does not need to be saved. In summary, CONDITIONAL BRANCH can be used as an unconditional branch provided the state of the carry flip-flop 76 is known to be reset (i.e., provided the conditional branch does not follow an arithmetic or an interrogation of pointer or status instruction).

ARITHMETIC & REGISTER (type 2) instructions apply to the arithmetic and register circuit 32 only. There are thirty-two ARITHMETIC & REGISTER instructions divided into eight classes encoded by the left most five bits of the instruction. Each of these instructions can be combined with any of eight word select signals to give a total capability of two hundred fifty-six instructions. The thirty-two ARITHMETIC & REGISTER instructions are listed in the table below.

TABLE OF TYPE TWO INSTRUCTIONS
(in order of binary code)

CODE	INST	CODE	INST
0 0000	0–B	1 0000	A–B
0 0001	0→B	1 0001	B←→C
0 0010	A–C	1 0010	SHIFT C RIGHT
0 0011	C–1	1 0011	A–1
0 0100	B→C	1 0100	SHIFT B RIGHT
0 0101	0–C→C	1 0101	C+C→C
0 0110	0→C	1 0110	SHIFT A RIGHT
0 0111	0–C–1→C	1 0111	0→A
0 1000	SHIFT A LEFT	1 1000	A–B→A
0 1001	A→B	1 1001	A←→B
0 1010	A–C→C	1 1010	A–C→A
0 1011	C–1→C	1 1011	A–1→A
0 1100	C→A	1 1100	A+B→A
0 1101	0–C	1 1101	A←→C
0 1110	A+C→C	1 1110	A+C→A
0 1111	C+1→C	1 1111	A+1→A

KEY: A, B, C, are registers, → means goes into ←→ means interchange.

The eight classes of ARITHMETIC & REGISTER instructions together with a parenthetical indication of the number of instructions within each class are as follows:

(1) CLEAR(3)
(2) TRANSFER/EXCHANGE(6)
(3) ADD/SUBTRACT (7)
(4) COMPARE(6)
(5) COMPLEMENT(2)
(6) INCREMENT(2)
(7) DECREMENT(2)
(8) SHIFT(4)

There are three CLEAR instructions. These instructions are $O \rightarrow A$, $O \rightarrow B$, and $O \rightarrow C$. They are implemented by simply disabling all the gates entering the designated register. Since these instructions can be combined with any of the eight word select options, it is possible to clear a portion of a register or a single digit.

There are six TRANSFER/EXCHANGE instructions. These instructions are $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow A$, $A \longleftrightarrow B$, $B \longleftrightarrow C$, and $C \longleftrightarrow A$. This variety permits data in registers A, B, and C to be manipulated in many ways. Again, the power of the instruction must be viewed in conjunction with with word select option. Single digits may be exchanged or transferred.

There are seven ADD/SUBTRACT instructions which use the adder circuity 82. They are $A \pm C \rightarrow C$, $A \pm B \rightarrow A$, $A \pm C \rightarrow A$, and $C+C \rightarrow C$. The last instruction can be used to divide by five. This is accomplished by first adding the number to itself via $C+C \rightarrow C$, then multiplying by two, then shifting right one digit, and dividing by ten. The result is a divide by five. The algorithm is used in the square root routine.

There are six COMPARE instructions. These instructions are always followed by a CONDITIONAL BRANCH. They are used to check the value of a register or a single digit in a register without modifying or transferring the contents thereof. These instructions (type 2) may easily be found in the instruction table above since there is no transfer arrow present. They are:

(1) 0—B (Compare B to zero)
(2) A—C (Compare A and C)
(3) C—1 (Compare C to one)
(4) 0—C (Compare C to zero)
(5) A—B (Compare A and B)
(6) A—1 (Compare A to one)

If, for example, it is desired to branch if B is zero (or any digit or group of digits is zero as determined by WS) the 0—B instruction is followed by a CONDITIONAL BRANCH. If B was zero, no carry (or borrow) would be generated and the branch would occur. Again, it is easy to compare single digits or a portion of a register by appropriate word select options.

There are two COMPLEMENT instructions. The number representation system in the calculator is sign and magnitude notation for the mantissa, and tens complement notation in the exponent field. Before numbers can be subtracted, the subtrahend must be tens-complemented (i.e., $0-C \rightarrow C$). Other algorithms require the nines complement (i.e., $0-C-1 \rightarrow C$).

There are four INCREMENT/DECREMENT instructions. They are $A \pm 1 \rightarrow A$ and $C \pm 1 \rightarrow C$.

There are four SHIFT instructions. All three registers A, B and C may be shifted right, while only A may be shifted left. The ARITHMETIC & REGISTER instrucion set is summarized by class in the table below.

TABLE OF TYPE TWO INSTRUCTIONS

(divided by class)

	CLASS	INSTRUCTION	CODE
1)	CLEAR	0→A	10111
		0→B	00001
		0→C	00110
2)	TRANSFER/ EXCHANGE	A→B	01001
		B→C	00100
		C→A	01100
		A←→B	11001
		B←→C	10001
		C←→A	11101
3)	ADD/SUBTRACT	A+C→C	01110
		A–C→C	01010
		A+B→A	11100
		A–B→A	11000
		A+C→A	11110
		A–C→A	11010
		C+C→A	10101
4)	COMPARE	0–B	00000
		0–C	01101
		A–C	00010
		A–B	10000
		A–1	10011
		C–1	00011
5)	COMPLEMENT	0–C→C	00101
		0–C–1→C	00111
6)	INCREMENT	A+1→A	11111
		C+1→C	01111
7)	DECREMENT	A–1→A	11011
		C–1→C	01011
8)	SHIFT	Sh A Right	10110
		Sh B Right	10100
		Sh C Right	10010
		Sh A Left	01000

The 28-bit shift register 68—72 in the control and timing circuit 30 contains twelve status bits or flags used to remember conditions of an algorithm or some past event (e.g., that the decimal point key has already been depressed). These flags may be individually set, reset, or interrogated or all bits may be cleared (reset simultaneously). The format for the STATUS OPERATION instructions (type 3) given in the instruction type table above is repeated below.

TABLE OF STATUS INSTRUCTION DECODING

BIT #	I_9 I_8 I_7 I_6	I_5 I_4	I_3	I_2 I_1 I_0
FIELD	N	F	0	1 0 0

F	INSTRUCTION
0 0	SET FLAG N
0 1	INTERROGATE FLAG N
1 0	RESET FLAG N
1 1	CLEAR ALL FLAGS (N = 0000)

If status bit N is one when the INTERROGATE N instruction is executed, the carry flip-flop 76 in the control and timing circuit 30 will be set. The status bit will remain set. INTERROGATE is always followed by a CONDITIONAL BRANCH instruction. The effective form of the INTERROGATE instruction is IF STATUS BIT N=0, THEN BRANCH or IF STATUS BIT N≠1, THEN BRANCH. The reason for this negative orientation is that all branches occur if the test is false (i.e., carry flip-flop=0), a result derived from combining conditional and unconditional branches under one instruction.

Status bit 0 is set when a key is depressed. If cleared it will be set every word time as long as the key is down.

A four-bit counter 54 in the control and timing circuit 30 functions as a pointer or marker to allow arithmetic instructions to operate on a portion of a register. Instructions are available to set and interrogate the pointer at one of fourteen locations or to increment or decrement the present position. The pointer instruction decoding is given in the table below.

TABLE OF POINTER INSTRUCTION DECODING

BIT #	9 8 7 6	5 4	3	2 1 0
FIELD	P	F	1	1 0 0

F	INSTRUCTION	
00	SET POINTER TO P	
10	INTERROGATE IF POINTER AT P	
01	DECREMENT POINTER	P = XXXX i.e., don't care
11	INCREMENT POINTER	

As with the STATUS INTERROGATE instruction, the carry flip-flop 76 is set if the pointer is at P when the INTERROGATE IF POINTER AT P instruction is executed. Like the status interrogation, the actual question is in the negative form IF P≠N, THEN BRANCH. This instruction would be followed by a CONDITIONAL BRANCH. In a maths routine the pointer allows progressive operation on a larger and larger portion of a word. After each iteration through a

loop, the pointer is decremented (or incremented) and then tested for completion to force another iteration or a jump out of the loop.

The DATA ENTRY & DISPLAY instructions (type 5) are used to enter data into the arithmetic and register circuit 32, manipulate the stack and memory registers, and blank the display. Sixteen instructions in this set are not recognized by any of the existing circuits and are, therefore, available for other external circuits that might be employed with other embodiments of the calculator. The table below contains a detailed listing of the DATA ENTRY & DISPLAY instructions.

TABLE OF TYPE 5 INSTRUCTIONS

(X = don't care, which in this context means the instruction does not depend on this bit; either a 1 or a 0 here will cause the same execution.)

$I_9\ I_8\ I_7\ I_6\ I_5\ I_4\ 1\ 0\ 0\ 0$

I_9 I_8 I_7 I_6	I_5 I_4	INSTRUCTION
0000 → 1111	0 0	16 Available Instructions
0000 → N 1001	0 1	Enters 4-bit code N into C Register at pointer position (LOAD CONSTANT)
0 0 0 0	1 X	DISPLAY TOGGLE
0 0 1 0	1 X	EXCHANGE MEMORY, C→M→C
0 1 0 0	1 X	UP STACK, C→C→D→E→F
0 1 1 0	1 X	DOWN STACK, F→F→E→D→A
1 0 0 0	1 X	DISPLAY OFF
1 0 1 0	1 X	RECALL MEMORY, M→M→C
1 1 0 0	1 X	ROTATE DOWN, C→F→E→D→C
1 1 1 0	1 X	CLEAR ALL REGISTERS 0→A,B,C,D,E,F,M
X X 0 1	1 X	I_S→A register (56 bits)
X X 1 1	1 X	BCD→C register (56 bits)

The first set of sixteen possible instructions (I_5I_4=00) in the table above is not used by the calculator.

The next instruction (I_5I_4=01) in this table is called the LOAD CONSTANT (LDC) or DIGIT ENTRY instruction. When this instruction is executed, four bits in I_9—I_6 will be inserted into the C-register at the location of the pointer, and the pointer will be decremented. This allows a constant, such as π (pi), to be stored in

read-only memory and then transferred to the arithmetic and register circuit 32. To transfer a ten-digit constant requires only eleven instructions (one to preset the pointer). Several exceptions exist in the use of this instruction. When used with the pointer in position 13, it cannot be followed by an ARITHMETIC & REGISTER instruction (i.e., type 2 or 5 instructions). With P=12, LOAD CONSTANT can be followed by another LOAD CONSTANT but not by any other type 2 or 5 instructions. When used with the pointer in position 14, the LOAD CONSTANT instruction has no effect. However, when P=12 and LOAD CONSTANT is followed by a type 2 or 5 instruction, position 13 in the C-register is modified. Loading non-digit codes (1010—1111) is not allowed since they will be modified when passing through the adder.

The next set of instructions (I_6 I_5 I_4=01X) in the type 5 instruction table above includes two DISPLAY instructions and six STACK or MEMORY instructions. The display flip-flop in the arithmetic and register circuit 32 controls blanking of all the LEDs. When it is reset, the 1111 code is set into the display buffer 94, which is decoded so that no segments are on. There is one instruction to reset this flip-flop (I_9 I_8 I_7=100) and another to toggle it (000). The toggle feature is convenient for blinking the display.

The remaining type 5 instructions include two affecting memory (EXCHANGE C⟷M and RECALL M→C), three affecting the stack (UP, DOWN, and ROTATE DOWN), one general CLEAR, one for loading register A from the line I_8 (I_7 I_6 I_5=011), and one for loading the C-register from the BCD line (111). Neither of the last two instructions depends on bits I_9, I_8, or I_4. The I_8→A instruction is designed to allow a key code to be transmitted from a program storage circuit to the arithmetic and register circuit 32 for display. The entire fifty-six bits are loaded, although only two digits of information are of interest. The BCD→C instruction allows data input to the arithmetic and register circuit 32 from a data storage circuit or the I/O circuit.

The ROM SELECT and other type six instructions are denoted by the pattern 10000 in instruction bits I_4—I_0. The decoding table for these instructions is shown below.

TABLE OF TYPE 6 INSTRUCTIONS

CIRCUIT AFFECTED	I_9 I_8 I_7	I_6 I_5	I_4 I_3 I_2 I_1 I_0	INSTRUCTION
ROM	0 0 0	0 0	1 0 0 0 0	ROM SELECT. ONE OF EIGHT AS
	↓	0 0	1 0 0 0 0	SPECIFIED IN BITS I_9 – I
	1 1 1	0 0	1 0 0 0 0	
CONTROL & TIMING	X X X	0 1	1 0 0 0 0	SUBROUTINE RETURN
	X X	0 1 0	1 0 0 0 0	EXTERNAL KEY CODE ENTRY TO CONTROL & TIMING CIRCUIT
	X X	1 1 0	1 0 0 0 0	KEYBOARD ENTRY
DATA STORAGE	1 X	0 1 1	1 0 0 0 0	SEND ADDRESS FROM C–REGISTER INTO DATA STORAGE CIRCUIT
	1 0	1 1 1	1 0 0 0 0	SEND DATA FROM C–REGISTER INTO DATA STORAGE CIRCUIT

The ROM SELECT instruction allows transfer of control from one ROM to another. Each ROM circuit has a masking option which is programmed to decode bits I_9—I_7. A SELECT ROM 3 instruction read from ROM circuit 1 will reset the ROE flip-flop 96 in ROM circuit 1 and set the ROE flip-flop 96 in ROM circuit 3. The address is incremented in the control and timing circuit 30 as usual. Thus, if

SELECT ROM 3 is in location 197 in ROM circuit 1, the first instruction read from ROM circuit 3 will be location 198.

Bits I_6I_5=01 designate a SUBROUTINE RETURN (RET) instruction. There are eight bits of storage in the 28-bit shift register 68—72 of the control and timing circuit 30 for retaining the return address when a JUMP SUBROUTINE instruction is executed. This address has already been incremented so execution of RET is simply a matter of outputting the address on the I_a line at the bit times b_{19}—b_{26} and also inserting it into the ROM address portion 68 of the shift register. It is also still retained in the return address portion 70.

A key code is entered into the control and timing circuit 30 by depressing a key on the keyboard. A key depression is detected by a positive interrogation of status bit 0. During a computation the keyboard is locked out because this status bit would ordinarily not be interrogated until a return to the display loop. The actual key depression saves the state of the system counter, which is also the key code, in the key code buffer 66 (see Figure 6) and also sets status bit 0. Execution of the KEYBOARD ENTRY instruction routes the key code (six bits) in the key code buffer 66 onto the I_a line and into the ROM address register 68 at bit times b_{19}—b_{24}. The most significant two bits b_{25} and b_{26} are set to zero so that a KEYBOARD ENTRY instruction always jumps to one of the first sixty-four states.

Two algorithms will now be discussed to further illustrate the instruction set. The first of these algorithms is a display wait loop which is employed after a key has been processed and while waiting for another key to be actuated. The second of these algorithms is a floating point multiply operation.

A flow diagram of the diplay wait loop is shown in Figure 43. This loop is entered after a keystroke has been processed, register A has been properly loaded with the number to be displayed, and register B contains the display mask as discussed above. Two flags or status bits are required. Status bit 0 (S0) is hardwired in the control and timing circuit 30 to be automatically set whenever a key is down. Status bit 8 (S8) is used in this program to denote the fact that the key which is presently down has already been processed since a routine may be finished before the key is released. In states DIS1 and DIS2, these two status bits are initialized. Then a loop is used as a time delay to wait out any key bounce. In DIS4, status bit 8 (S8) is checked. The first time through the algorithm is must be 1 since it was set in DIS1 to indicate that the key has been processed. In state DIS5 the display is turned on. Actually, the display is toggled since it must previously have been off. There is no separate instruction for turning on the display. At this time the answer appears to the user. In DIS6, status bit 0 (S0) is checked to see if a key is down. If not (i.e., S0=0), the previous key has been released and status bit 8 (S8) is reset to 0 (DIS7). The machine is now ready to accept a new key since the previous key has been processed and released. The algorithm cycles through DIS6 and DIS7 waiting for a new key. This is the basic wait cycle of the calculator. If SO=1 in the DIS6, the key which is down may be the old key (i.e., the one just processed) or a new key. This can be determined upon return to DIS4 where status bit 8 (S8) is checked. If a new key is down (S8=0), execution jumps to DIS8, the display is blanked, and a jump out is made to service the key. A listing of the algorithm is given in the table below.

TABLE OF WAIT LOOP ALGORITHM

LABEL	OPERATION	COMMENT
DIS1:	1 → S8	SET STATUS 8
DIS2:	0 → S0	RESET STATUS 0
DIS3:	P←1 → P	DECREMENT POINTER,
	IF P # 12	48 WORD LOOP (3 × 16)
	THEN GO TO DIS3	TO WAIT OUT KEY BOUNCE
	DISPLAY OFF	
DIS4:	IF S8 # 1	IF KEY NOT PROCESSED,
	THEN GO TO DIS8	LEAVE ROUTINE
DIS5:	DISPLAY TOGGLE	TURN ON DISPLAY
DIS6:	IF SØ # 1	IF KEY UP, RESET
	THEN GO TO DIS7:	S8 AND WAIT
	GO TO DIS2:	KEY DOWN. CHECK IF SAME KEY
DIS7:	0 → S8	INDICATE KEY NOT PROCESSED
	GO TO DIS6:	BACK TO WAIT FOR KEY
DIS8:		BLANK DISPLAY
DIS9:	KEYS → ROM ADDRESS ↓ CONTINUE	JUMP TO START OF PROGRAM TO PROCESS KEY THAT WAS DOWN

The floating point multiply algorithm multiplies x times y, where the register C contains x in scientific notation and the register D contains y. When the multiply key is depressed the wait loop algorithm will jump to a ROM address corresponding to the first step of multiply algorithm because of the way the instruction KEYS→ROM ADDRESS (state DIS9 in Fig. 43) is executed. The key code actually becomes the next ROM address. At this time the contents of registers A—D are indicated by the following:

Register A contains the floating point form of x
Register B contains the display mask for x
Register C contains the scientific form of x
Register D contains the scientific form of y

The algorithm for executing floating multiply is given in the table below. The letters in parentheses indicate word select options as follows:

P = POINTER POSITION
WP = UP TO POINTER POSITION
X = EXPONENT FIELD
XS = EXPONENT SIGN
M = MANTISSA FIELD WITHOUT SIGN
MS = MANTISSA WITH SIGN
W = ENTIRE WORD
S = MANTISSA SIGN ONLY

TABLE OF FLOATING POINT MULTIPLY ALGORITHM

LABEL	OPERATION	COMMENT
MPY1:	STACK → A	TRANSFER y TO A. DROP STACK
MPY2:	A+C → C(X)	ADD EXPONENTS TO FORM EXPONENT OF ANSWER.
	A+C → C(S)	ADD SIGNS TO FORM SIGN
	IF NO CARRY GO TO MPY3	OF ANSWER
	0 → C(S)	CORRECT SIGN IF BOTH NEGATIVE,
MPY3:	0 → B(W)	CLEAR B, THEN TRANSFER
	A → B(M)	MANTISSA OF y.
	0 → A(W)	PREPARE A TO ACCUMULATE PRODUCT.
	2 → P	SET POINTER TO LSD (LEAST SIGNIFICANT DIGIT) MULTIPLIER (MINUS 1).
MPY4:	P+1 → P	INCREMENT TO NEXT DIGIT.
MPY5:	A+B → A(W)	ADD MULTIPLIER MANTISSA TO PARTIAL
	C−1 → C(P)	PRODUCT C(P) TIMES. WHEN C(P) = 0,
	IF NO CARRY GO TO MPY5	STOP AND GO TO NEXT DIGIT.
	SHIFT RIGHT A(W)	SHIFT PARTIAL PRODUCT RIGHT.
	IF P # 12	CHECK IF MULTIPLY IS COMPLETE
	THEN GO TO MPY4	(IS POINTER AT MSD)
	IF A(P) > 1	CHECK IF MSD = 0. IF SO MUST
	THEN GO TO MPY6	SHIFT LEFT AND CORRECT EXPONENT.
	SHIFT LEFT A(M)	MULTIPLY BY 10 AND DECREMENT EXPONENT.
	C−1 → C(X)	
MPY6:	C+1 → C(X)	ALWAYS DO THIS TO CORRECT FOR FACTOR OF 10 TOO SMALL.
	A → B(XS)	DUPLICATE EXTRA PRODUCT DIGITS
	A+B → A(XS)	ADD 11th DIGITS.
	IF NO CARRY GO TO MPY7	IF SUM LESS THAN 10, THEN DONE.
	A+1 → A(M)	IF SUM MORE THAN 10, ADD 1.
	IF NO CARRY GO TO MPY7	IF ANSWER WAS NOT ALL 9s, THEN DONE
	A+1 → A(P)	IF ANSWER WAS ALL 9s ADD 1
	C+1 → C(X)	AND INCREMENT EXPONENT.
MPY7:	A EXCHANGE C(M)	GET ANSWER MANTISSA INTO C.
	GO TO MASK 1	GO TO ROUTINE TO POSITION THE ANSWER IN A AND MAKE THE PROPER MASK IN B.

INSTRUCTION SET 2

Whereas the instructions contained within the instruction set 1 are those executed by the arithmetic and register circuit, control and timing circuit, read-only memory circuits, and data storage circuit, the instruction set 2 comprises those instructions executed by the input/output (I/O) circuit.

In some instances the same bit pattern may be representative of an instruction within the instruction set 1 as well as an instruction within the instruction set 2. This situation requires that the various calculator circuits be made aware of the instruction set under which an instruction is given. In view of this requirement, an IS1 or IS2 mode instruction is issued to designate that instructions which follow are to be taken from either instruction set 1 or instruction set 2, respectively. Following the execution of an IS2 instruction, the arithmetic and register circuit and the control and timing circuit are inhibited from responding to subsequent instructions, since they will be instruction set 2 types.

The instructions comprising instruction set 2 are described in the following table, with bit patterns shown in ascending order of significance from right to left.

TABLE OF INSTRUCTIONS (INSTRUCTION SET 2)

MNEMONIC	BIT PATTERN	DESCRIPTION
ISL	1001000000	ENABLE INSTRUCTION SET 1
IS2	0101000000	ENABLE INSTRUCTION SET 2
TTC	0111111000	T–REGISTER → C–REGISTER
CTT	0111110000	C–REGISTER → T–REGISTER
XOR	0010001000	T–REGISTER EXCLUSIVE OR C–REGISTER → T–REGISTER
IOR	0011001000	T–REGISTER INCLUSIVE OR C–REGISTER → T–REGISTER
AND	0010101000	T–REGISTER LOGICAL AND C–REGISTER → T–REGISTER
ADD	0001101000	T–REGISTER LOGICAL AND C–REGISTER → T–REGISTER
SLT	0100001000	SHIFT T–REGISTER LEFT ONE BIT
SRT	0101001000	SHIFT T–REGISTER RIGHT ONE BIT
TINC	1100001000	INCREMENT T–REGISTER BY ONE
TDEC	1101001000	DECREMENT T–REGISTER BY ONE
YBC	1110001000	INTERROGATE BINARY CARRY
EERA	0001010000	EXTERNAL ENTRY TO ROM A
IXT	0100101000	I/O REGISTER ⇐→ REGISTER
CCS	1111110000	COMPARE C–REGISTER WITH SECTOR COUNTER
TCS	1101110000	COMPARE T–REGISTER WITH SECTOR COUNTER
PRE	0101110000	PRINTER ENABLE
ADV	0011110000	ADVANCE PAPER
RED	0001110000	PREPARE FOR RED PRINT
TTP	1000000000	LOAD T–REGISTER INTO PROGRAM COUNTER
PTT	1000101000	LOAD PROGRAM COUNTER INTO T–REGISTER

MNEMONIC	BIT PATTERN	DESCRIPTION
PINC	0111000000	INCREMENT PROGRAM COUNTER BY ONE
PDEC	1001001000	DECREMENT PROGRAM COUNTER BY ONE
RMGRA	0000100000	SELECT ROM GROUP A
RMGRB	0010100000	SELECT ROM GROUP B
RMGRC	0100100000	SELECT ROM GROUP C
YADP	0110100000	INTERROGATE AUTO DECIMAL POINT SWITCH
READ	1000100000	READ PROGRAM MEMORY
SBL	1010100000	SET BUSY LIGHT
YFKB	1100100000	INTERROGATE FUNCTION BLOCK KEYBOARD FLAG
YPOC	1110100000	INTERROGATE PRINT-ON COMMAND FLAG
YINTF	0000100000	INTERROGATE FLAG
WRITE	0010100000	WRITE PROGRAM INSTRUCTION
RBL	0100100000	RESET BUSY LIGHT
GIOE	0110100000	GENERAL I/O ENABLE
ELON	1010100000	ERROR LIGHT ON
TG9	1100100000	GENERAL I/O INSTRUCTION 1
TG8	1110100000	GENERAL I/O INSTRUCTION 2

SYSTEM MICROPROGRAMMING

Referring to Figure 42, there is shown a flow chart of the overall calculator microprogramming. A detailed listing of the routines shown in this diagram is given later in this specification. The start-up sequence begins by placing the calculator power switch in the "on" position. This forces the control and timing circuit to enable ROM group A, the basic calculator ROM group 34, and to activate address 000 on ROM circuit A—Ø. The instruction located at this address commences execution of the machine initiation routine. The purposes of this routine are to clear all working registers, clear the status bits register, place the calculator in the manual control mode, set the numeric output format to fixed point with two decimal places, call the initiation routines associated with peripheral I/O units, and indicate the status of the calculator system by printing CLEAR on the ouput printer unit.

Upon completion of the machine initiation routine, control is passed to the system supervisor routines. The first tests are routine user code and operation mode code. A nonzero routine user code causes control of the system to be transferred to the keyboard function block routines issuing ROM group B call

instructions. If the result of the user code test is zero, the operation mode test is activated. A zero result indicates manual control mode, whereas a nonzero result indicates the program run mode.

In the manual control mode, the contents of the C-register in the arithmetic and register circuit are formatted for numeric output, the appropriate display mask is built, and the display is switched on. A nonzero data storage user flag results in control being transferred to the data storage routines. If this flag test is zero, control remains in the main system. If the key previously depressed is in the down position, the calculator system remains in a wait loop until that key is released. Releasing a key at this point indicates completion of the previous execution routine and forces the calculator into an idle loop, waiting for a new key to be depressed.

When a new key is depressed, a special flag is set in the status bit register which causes the calculator to exit the wait loop. After approximately 3 milliseconds of delay the central processor is prepared for execution of a new routine by resetting some pointer flags and by properly allocating data to various registers. The short time delay is used to eliminate any problems associated with bouncing keyboard switch contacts. If the currently depressed key is associated with a keyboard function block, control is transferred to the function block routines. If the currently depressed key is preceded by the SHIFT key, control is then transferred to ROM circuit A—6 where execution routines for insructions from ROM circuit A—2 and ROM group C have their beginning points. Otherwise, the central processor is prepared to accept a digit entry by resetting the A-register. Execution routines contained in ROM circuit A—1 are now called.

In the program run mode, a test is first made to determine if the STOP key has been depressed. If it has, the proper flags are changed to indicate a mode change from program run to manual control, after which control is transferred to the system supervisor starting point. If the STOP key has not been depressed, the program counter is incremented, and a new instruction is read from program memory. If the program memory has not been exceeded, the central processor is prepared to accept a digit entry and to accept a new instruction as the starting address for a routine. In the event memory capacity is exceeded during an increment step, an error routine is called which results in an error message being printed. At the same time, the calculator is returned to manual control and microprogramming control is transferred to the beginning point of the system supervisor.

Each execution routine is associated with a particular instruction from the machine language. All of these routines are described in detail below in that portion of the instruction listings which deals with ROM circuits A—Ø to A—7. The routine associated with the left parenthesis key is shown in flow chart form at Figure 48.

DETAILED LISTING OF ROUTINES AND SUBROUTINES OF INSTRUCTIONS

A complete listing of all of the routines and subroutines of instructions employed by the calculator and of all of the constants employed by these routines and subroutines is given below. The listing includes all routines and subroutines employed by the calculator as basically and optionally configured. All of these routines, subroutines, and constants are stored in the individual ROM circuits within each ROM group, as indicated at the top of the first page associated with each ROM group. Each line of listing associated with each ROM circuit is separately numbered in the first column from the left-hand side of each page. This facilitates reference to different parts of the listing. Each address within each of the ROM circuits is represented in octal form by four digits in the second column from the left-hand side of each page. Branching addresses are represented in octal form by four digits in the fourth column from the left-hand side of each page. The bit pattern of each instruction or constant is represented as ten bits in the fourth column from the left-hand side of each page. Labels associated with particular ones of the instructions are located in the fifth column from the left-hand side of each page. A mnemonic representation for each of the instructions is shown in the sixth column from the left-hand side of each page. Labels associated with the branch instructions are represented in the seventh column from the left-hand side of each page. Explanatory comments are given in the remaining portion of each page.

MAIN UNIT LISTING (ROM A—Ø)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		0 000 110 100	LPWO1	CLS			CLEAR ALL STATUS BITS
4	0001	0114	0 100 110 101		JSB	LPWO2		GO TO I/O SOFTWARE PWON
5	0002		1 110 101 000	LPWO3	CLR			CLEAR ALL REGISTERS
6	0003		0 010 100 000		RMGRB			CALL I.BLK SOFTWARE FOR REDEFINITION
7	0004		1 101 001 100		PT13			PREPARE DP FLAG LOAD
8	0005	0010	0 000 100 011		BRN	LPOW4		SET IT TO 2
9	0006		0 000 101 000	LSRT2	DSTO			SWITCH THE DISPLAY ON
10	0007		1 010 101 000		MTC			PREPARE INTERNAL FLAG VECTOR
11	0010		0 010 011 000	LPOW4	LDC2			LOAD DP FLAG OR NONZERO OPMODE FLAG
12	0011		0 010 010 000	LRES3	ROM 1			
13	0012	0213	1 000 101 111	LDIV1	BRN	LDIV2		GO TO DIVIDE ROUTINE
14	0013	0210	1 000 100 011	LMUL1	BRN	LMUL2		GO TO MULTIPLICATION ROUTINE
15	0014	0163	0 111 010 001	LPOW1	JSB	LPRS1		PREPARE DATA AND BLANK FOR PRINT
16	0015	0205	1 000 010 111		BRN	LPOW2		BRANCH TO CONTINUE
17	0016		1 000 101 110	LRIP1	W,BXC			SAVE DATA TEMPORARILY IN B-REG
18	0017	0031	0 001 100 111		BRN	LRIP2		BRANCH TO CONTINUE
19	0020	0163	0 111 010 001	LEQL1	JSB	LPRS1		PREPARE DATA AND BLANK FOR PRINT
20	0021		0 110 010 000		ROM 3			
21	0022		0 101 000 100	LDEC1	SS5			DECIMAL POINT ENTRY, SET D.P. KEY FLAG
22	0023		1 000 101 110	LDIG0	W,BXC			DIGIT Ø ENTRY, STORE DATA IN B-REG
23	0024		1 010 101 000		MTC			PREPARE INT. FLG. VECTOR FOR TEST
24	0025	0071	0 011 100 111		BRN	LDGT1		PREPARE ROM SWITCH
25	0026		0 110 010 000	LPAR1	ROM 3			
26	0027	0176	0 111 111 101	LRESC	JSB	LPRTØ		JUMP TO FORCED PRINT ROUTINE
27	0030		0 010 010 000		ROM 1			SWITCH TO ROM 1 AGAIN
28	0031		0 100 010 000	LRIP2	ROM 2			SWITCH TO ROM 2
29	0032		1 111 101 010	LDIG3	X,AP1A			DIGIT 3 ENTRY
30	0033		1 111 101 010	LDIG2	X,AP1A			DIGIT 2 ENTRY
31	0034		1 111 101 010	LDIG1	X,AP1A			DIGIT 1 ENTRY
32	0035	0023	0 001 001 111		BRN	LDIGØ		GO TO DIGITØ ENTRY
33	0036	0146	0 110 011 011	LPER1	BRN	LPERA		PERCENT ROUTINE BEGINNING POINT
34	0037		0 110 010 000	LPERY	ROM 3			
35	0040	0163	0 111 010 001	LADD1	JSB	LPRS1		PREPARE DATA AND BLANK FOR PRINT
36	0041	0202	1 000 001 011		BRN	LADD2		BRANCH TO CONTINUE
37	0042		1 111 101 010	LDIG6	X,AP1A			DIGIT 6 ENTRY
38	0043		1 111 101 010	LDIG5	X,AP1A			DIGIT 5 ENTRY
39	0044		1 111 101 010	LDIG4	X,AP1A			DIGIT 4 ENTRY
40	0045	0032	0 001 101 011		BRN	LDIG3		GO TO DIGIT 3 ENTRY
41	0046	0061	0 011 000 111	LMTX1	BRN	LXTMIØ		RECALL, GO TO COMMON PART WITH STORE
42	0047		0 010 100 000	LFBLK1	RMGRB			SWITCH TO ROM GROUP 3
43	0050	0163	0 111 010 001	LSUB1	JSB	LPRS1		PREPARE DATA AND BLANK FOR PRINT
44	0051	0344	1 110 010 011		BRN	LSUB2		BRANCH TO CONTINUE
45	0052		1 111 101 010	LDIG9	X,AP1A			DIGIT 9 ENTRY
46	0053		1 111 101 010	LDIG8	X,AP1A			DIGIT 8 ENTRY
47	0054		1 111 101 010	LDIG7	X,AP1A			DIGIT 7 ENTRY
48	0055	0042	0 010 001 011		BRN	LDIG6		GO TO DIGIT 6 ENTRY
49	0056	0163	0 111 010 001	LXTM1	JSB	LPRS1		PREPARE DATA NAD BLANK FOR RPINT
50	0057	0173	0 111 110 001		JSB	LPRT3		PREPARE SYMBOL AND PRINT
51	0060		1 001 000 100		SS9			SET "STORE" ROUTINE FLAG
52	0061		0 100 010 000	LXTMIØ	ROM 2			

MAIN UNIT LISTING (ROM A—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
53	0062	0271	1 011 100 111	LCONT1	BRN	LCONT2			
54	0063	0321	1 101 000 111	LENTR	BRN	LSLE1	LAST ENTRY KEYCODE ACCEPTED		
55	0064	0355	1 110 111 001	LSRT1	JSB	LKYUP1	WAIT FOR KEY RELEASE		
56	0065	0006	0 000 011 011		BRN	LSRT2			
57	0066	0260	1 011 000 011	LCEN1	BRN	LCEN2			
58	0067		0 010 010 000	LRND2	ROM 1		SWITCH TO ROM 1		
59	0070		1 100 010 000	LSHFT1	ROM 6				
60	0071		0 010 010 000	LDGT1	ROM 1		SWITCH TO ROM 1		
61	0072	0067	0 011 011 111	LRND1	BRN	LRND2	ROUND KEY ROUTINE BEGINNING		
62	0073		0 100 000 100	LRES1	SS4		SET AUX FLAG TO INDICATE RESET ROUTINE		
63	0074		1 011 001 100	LRES2	PT11		SET POINTER TO AVOID OP.MOBE CHANGE		
64	0075	0162	0 111 001 011		BRN	LRES8			
65	0076	0222	1 001 001 101	LPRX1	JSB	LPRX2	STORE RETURN VECTOR		
66	0077	0165	0 111 011 001		JSB	LPRT9	AFTER RETURN LOAD SYMBOL # AND PRINT		
67	0100		0 010 010 000		ROM 1		GO TO ROUTINE EXIT		
68	0101		1 010 101 000	LSUP10	MTC		PREPARE INT. FLG. VECTOR		
69	0102		1 110 101 110		W,AXC		RESTORE DATA IN C-REG		
70	0103		0 111 001 100		PT7		PREPARE ROUTINE USER CODE TEST		
71	0104		1 101 100 010		P,AM1A		IS IT ZERO?		
72	0105	0047	0 010 011 111		BRN	LFBLK1	NO; GO TO FUNCTION BLOCK		
73	0106	0111	0 100 100 111		BRN	LSUP2	YES; STAY IN MAIN UNIT		
74	0107		0 110 001 110	LSUP1	W,CTA				
75	0110	0101	0 100 000 111		BRN	LSUP10			
76	0111		1 100 001 100	LSUP2	PT12		PREPARE OP. MODE TEST		
77	0112		1 101 100 010		P,AM1A		IS IT ZERO?		
78	0113	0261	1 011 000 111		BRN	LRUN1	NO; GO TO PROGRAM RUN DIRECTOR		
79	0114	0147	0 110 011 111		BRN	LMAN1	YES; GO TO MANUAL CONTROL DIRECTOR		
80	0115		0 101 000 000	LPWO2	IS2				
81	0116	0037	0 010 000 001		LIO	100	LOAD CHANNEL 1 SELECT CODE		
82	0117		0 110 100 000		GIOE		I/O CHANNEL CALL		
83	0120		0 101 000 000		IS2				
84	0121	0077	0 100 000 001		LIO	200	LOAD CHANNEL 2 SELECT CODE		
85	0122		0 110 100 000		GIOE		I/O CHANNEL CALL		
86	0123		0 101 000 000		IS2				
87	0124	0177	1 000 000 001		LIO	400	LOAD CHANNEL 3 SELECT CODE		
88	0125		0 110 100 000		GIOE		I/O CHANNEL CALL		
89	0126		1 001 000 000		IS1				
90	0127	0141	0 110 000 111		BRN	LPWO8			
91	0130	0367	1 111 011 111	LEQLF	BRN	LEQLG			
92	0131		1 100 010 000	LPCHS	ROM 6		CHANGE SIGN ROUT.BEGINNING		
93	0132		1 000 101 110	LRIP8	W,BXC		SAVE DATA TEMPORARILY IN B-REG		
94	0133		0 110 010 000		ROM 3				
95	0134		0 101 000 000	LPAGE	IS2		CALL NEXT PAGE ROUTINE BEGINNING		
96	0135	0007	0 000 100 001		LIO	020	SET PROPER SELECT CODE		
97	0136		0 110 100 000		GIOE		ACTIVATE PAGING HARDWARE		
98	0137		1 001 000 000		IS1				
99	0140	0107	0 100 011 111		BRN	LSUP1			
100	0141		1 011 010 100	LPWO8	YS11		IS I/O FLAG SET ?		
101	0142	0002	0 000 001 011		BRN	LPWO3	NO,CONTINUE IN PWON ROUTINE		
102	0143	0273	1 011 101 111		BRN	LTRB1	I/O DEVICE NOT FOUND,GO TO ERROR ROUTINE		
103	0144	0163	0 111 010 001	LPERZ	JSB	LPRS1			
104	0145	0037	0 001 111 111		BRN	LPERY			
105	0146		0 100 010 000	LPERA	ROM 2				
106	0147		0 100 010 000	LMAN1	ROM 2		SWITCH TO ROM 2		
107	0150	0152	0 110 101 011	LJMPB	BRN	LJMPR	JUMP RELATIVE BACKWARD		
108	0151		0 100 010 000	LSCOT	ROM 2		ALLOWS TO REACH BEHIND DSP. ROUTINE FROM OUTSIDE		
109	0152		0 100 000 100	LJMPR	SS4		JUMP RELATIVE FORWARD		
110	0153	0227	1 001 011 111		BRN	LJMP1			
111	0154		0 111 101 010	LCOD5	X,CP1C		START TO GENERATE 5 FOR SECOND OP. CODE DIGIT		
112	0155		0 111 101 010	LCOD4	X,CP1C		START TO GENERATE 4 FOR SECOND OP. CODE DIGIT		

MAIN UNIT LISTING (ROM A—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	Ø156		Ø 111 1Ø1 Ø1Ø	LCOD3	X,CP1C	START TO GENERATE 3 FOR SECOND OP. CODE DIGIT
114	Ø157		Ø 111 1Ø1 Ø1Ø	LCOD2	X,CP1C	START TO GENERATE 2 FOR SECODE OP. CODE DIGIT
115	Ø160		Ø 111 1Ø1 Ø1Ø	LCOD1	X,CP1C	START GENERATE 1 FOR SECOND OP. CODE DIGIT
116	Ø161		1 ØØØ Ø1Ø ØØØ	LCOD0	ROM 4	USE ZERO FOR SECOND OP. CODE DIGIT
117	Ø162		1 Ø1Ø 1Ø1 ØØØ	LRES8	MTC	GET INT.FLG.VECTOR FOR UPDATE
118	Ø163	ØØ11	Ø ØØØ 1ØØ 111		BRN	LRES3
119	Ø164		1 11Ø Ø1Ø ØØØ	LPRS1	ROM 7	ROUTINE BEGINNING FOR OUTPUT "BLANK"
120	Ø165		Ø ØØØ 11Ø ØØØ	LPRSB	RETURN	END OF THE SUBROUTINE (ORIG.DATA IS IS A-REG)
121	Ø166		Ø 111 1ØØ Ø1Ø	LPRT9	P,CP1C	LOAD PRINT CODE FOR SYMBOL ≠
122	Ø167		Ø 111 1ØØ Ø1Ø	LPRT8	P,CP1C	LOAD PRINT CODE FOR SYMBOL
123	Ø170		Ø 111 1ØØ Ø1Ø	LPRT7	P,CP1C	LOAD PRINT CODE FOR SYMBOL
124	Ø171		Ø 111 1ØØ Ø1Ø	LPRT6	P,CP1C	LOAD PRINT CODE FOR SYMBOL
125	Ø172		Ø 111 1ØØ Ø1Ø	LPRT5	P,CP1C	LOAD PRINT CODE FOR SYMBOL
126	Ø173		Ø 111 1ØØ Ø1Ø	LPRT4	P,CP1C	LOAD PRINT CODE FOR SYMBOL <
127	Ø174		Ø 111 1ØØ Ø1Ø	LPRT3	P,CP1C	LOAD PRINT CODE FOR SYMBOL >
128	Ø175		Ø 111 1ØØ Ø1Ø	LPRT2	P,CP1C	LOAD PRINT CODE FOR SYMBOL =
129	Ø176		Ø 111 1ØØ Ø1Ø	LPRT1	P,CP1C	LOAD PRINT CODE FOR SYMBOL ×
130	Ø177		1 11Ø Ø1Ø ØØØ	LPRT0	ROM 7	LOAD PRINT CODE FOR SYMBOL +
131	Ø200		1 ØØØ Ø1Ø ØØØ	LINTE2	ROM 4	
132	Ø201		1 ØØØ Ø1Ø ØØØ	LYLE2	ROM 4	
133	Ø202	Ø176	Ø 111 111 1Ø1	LADD2	JSB	LPRT0 PREPARE SYMBOL + AND PRINT
134	Ø203	Ø160	Ø 111 ØØØ Ø11		BRN	LCOD1 GO TO LOAD SECOND OP. CODE DIGIT
135	Ø204	Ø144	Ø 11Ø Ø1Ø Ø11		BRN	LPERZ
136	Ø205	Ø171	Ø 111 1Ø1 ØØ1	LPOW2	JSB	LPRT5 PREPARE SYMBOL AND PRINT
137	Ø206	Ø154	Ø 11Ø 11Ø Ø11		BRN	LCOD5 GO TO LOAD SECOND OP. CODE DIGIT
138	Ø207		Ø 11Ø Ø1Ø ØØØ	LLABL	ROM 3	LABEL HEAD ROUTINE BEGINNING [DEC=87
139	Ø210	Ø163	Ø 111 Ø1Ø ØØ1	LMUL2	JSB	LPRS1 PREPARE DATA AND BLANK FOR PRINT
140	Ø211	Ø175	Ø 111 111 ØØ1		JSB	LPRT1 PREPARE SYMBOL × AND PRINT
141	Ø212	Ø156	Ø 11Ø 111 Ø11		BRN	LCOD3 GO TO LOAD SECOND OP. CODE DIGIT
142	Ø213	Ø163	Ø 111 Ø1Ø ØØ1	LDIV2	JSB	LPRS1 PREPARE DATA AND BLANK FOR PRINT
143	Ø214	Ø170	Ø 111 1ØØ 1Ø1		JSB	LPRT6 PREPARE SYMBOL AND PRINT
144	Ø215	Ø155	Ø 11Ø 11Ø 111		BRN	LCOD4 GO TO LOAD SECOND OP. CODE DIGIT
145	Ø216	Ø176	Ø 111 111 1Ø1	LPERB	JSB	LPRT0 GO TO PRINT SUBROUTINE
146	Ø217		Ø 11Ø Ø1Ø ØØØ	LPERQ	ROM 3	
147	Ø220		1 Ø1Ø 1ØØ ØØØ	LPRTZ	SBL	SET BUSY LIGHT
148	Ø221		Ø ØØØ 11Ø ØØØ	LPRTB	RETURN	END OF THE SUBROUTINE
149	Ø222	Ø132	Ø 1Ø1 1Ø1 Ø11	LRIPC	BRN	LRIP8
150	Ø223		1 11Ø Ø1Ø ØØØ	LPRX2	ROM 7	SWITCH TO ROM7 PRINT SUBROUTINES
151	Ø224		1 ØØ1 ØØØ 1ØØ	LJSBC	SS9	JUMP TO SUBROUTINE; LABEL IN C-REG
152	Ø225		1 ØØØ ØØØ 1ØØ	LJSBP	SS8	JUMP TO SUBROUTINE; LABEL IN PROGRAM
153	Ø226		Ø ØØ1 ØØØ 1ØØ	LJSBA	SS1	JUMP TO SUBROUTINE; GIVEN BY ADDRESS
154	Ø227		Ø 1Ø1 ØØØ 1ØØ	LJMP1	SS5	SET COMMON JUMP FLAG
155	Ø230		1 1ØØ Ø1Ø ØØØ	LJMPS	ROM 6	COMMON START FOR PROGRAM JUMPS
156	Ø231	Ø163	Ø 111 Ø1Ø ØØ1	LMTXB	JSB	LPRS1 PREPARE DATA AND BLANK FOR PRINT
157	Ø232	Ø172	Ø 111 1Ø1 1Ø1		JSB	LPRT4 PREPARE SYMBOL AND PRINT
158	Ø233		Ø Ø1Ø Ø1Ø ØØØ		ROM 1	ROUTINE EXIT
159	Ø234	Ø176	Ø 111 111 1Ø1	LEQL2	JSB	LPRT0 GO TO PRINT
160	Ø235		Ø 11Ø Ø1Ø ØØØ		ROM 3	
161	Ø236		Ø 11Ø Ø1Ø ØØØ	LGRND	ROM 3	GRAND TOTAL ROUTINE BEGINNING
162	Ø237		Ø Ø1Ø 1ØØ ØØØ	LTABP	RMGRB	TABLE FUNCTION IN PRGM.RUN MODE
163	Ø240		Ø Ø1Ø 1ØØ ØØØ	LFBC01	RMGRB	F.B. FUNCTION Ø1 BEGINNING
164	Ø241		Ø Ø1Ø 1ØØ ØØØ	LFBC02	RMGRB	F.B. FUNCTION Ø2 BEGINNING
165	Ø242		Ø Ø1Ø 1ØØ ØØØ	LFBC03	RMGRB	F.B. FUNCTION Ø3 BEGINNING
166	Ø243		Ø Ø1Ø 1ØØ ØØØ	LFBC04	RMGRB	F.B. FUNCTION Ø4 BEGINNING
167	Ø244		Ø Ø1Ø 1ØØ ØØØ	LFBC05	RMGRB	F.B. FUNCTION Ø5 BEGINNING
168	Ø245		Ø Ø1Ø 1ØØ ØØØ	LFBC06	RMGRB	F.B. FUNCTION Ø6 BEGINNING
169	Ø246		Ø Ø1Ø 1ØØ ØØØ	LFBC07	RMGRB	F.B. FUNCTION Ø7 BEGINNING
170	Ø247		Ø Ø1Ø 1ØØ ØØØ	LFBC08	RMGRB	F.B. FUNCTION Ø8 BEGINNING
171	Ø250		Ø Ø1Ø 1ØØ ØØØ	LFBC09	RMGRB	F.B. FUNCTION Ø9 BEGINNING
172	Ø251		Ø Ø1Ø 1ØØ ØØØ	LFBC1Ø	RMGRB	F.B. FUNCTION 1Ø BEGINNING

MAIN UNIT LISTING (ROM A—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	Ø252		Ø Ø1Ø 1ØØ ØØØ	LFBC11	RMGRB		F.B. FUNCTION 11 BEGINNING
174	Ø253		Ø Ø1Ø 1ØØ ØØØ	LFBC12	RMGRB		F.B. FUNCTION 12 BEGINNING
175	Ø254		Ø Ø1Ø 1ØØ ØØØ	LFBC13	RMGRB		F.B. FUNCTION 13 BEGINNING
176	Ø255		Ø Ø1Ø 1ØØ ØØØ	LFBC14	RMGRB		F.B. FUNCTION 14 BEGINNING
177	Ø256		Ø Ø1Ø 1ØØ ØØØ	LFBC15	RMGRB		F.B. FUNCTION 15 BEGINNING
178	Ø257	Ø3Ø5	1 1ØØ Ø1Ø 111		BRN	LERRA	F.B.SOFTWARE NOT READY
179	Ø26Ø		Ø Ø1Ø Ø1Ø ØØØ	LCEN2	ROM 1		
18Ø	Ø261		Ø ØØØ Ø1Ø 1ØØ	LRUN1	YSØ		IS ANY KEY DOWN?
181	Ø262	Ø3Ø1	1 1ØØ ØØØ 111		BRN	LRUN2	NO; CONTINUE IN PROGRAM RUN MODE
182	Ø263		Ø Ø1Ø 1Ø1 ØØØ	LSTOP	CXM		STOP ROUTINE BEGINNING,GET INT.FLG. VECTOR
183	Ø264		Ø Ø11 ØØØ Ø1Ø		P,ZTC		SET MANUAL CONTROL FLAG
184	Ø265		Ø Ø1Ø 1Ø1 ØØØ	LSTP1	CXM		RESTORE DATA,STORE NEW INT.FLG.VECTOR
185	Ø266	Ø1Ø7	Ø 1ØØ Ø11 111		BRN	LSUP1	GO TO SUPERVISOR
186	Ø267		1 1ØØ Ø1Ø ØØØ	LOOXS	ROM 6		1/X ROUTINE
187	Ø27Ø		1 1ØØ Ø1Ø ØØØ	LEXPS	ROM 6		EXP ROUTINE
188	Ø271	Ø355	1 11Ø 111 ØØ1	LCONT2	JSB	LKYUP1	WAIT FOR KEY RELEASE
189	Ø272		Ø Ø1Ø Ø1Ø ØØØ		ROM 1		
19Ø	Ø273		1 Ø11 1ØØ 1ØØ	LTRB1	RS11		RESET EXTERNAL FLAG
191	Ø274		1 11Ø Ø1Ø ØØØ		ROM 7		GO TO PRINT MESSAGE
192	Ø275		Ø ØØ1 1ØØ 11Ø	LIFZ1	M,CM1		CHECK VALUE OF MANTISSA
193	Ø276	Ø2ØØ	1 ØØØ ØØØ Ø11		BRN	LINTE2	NONZERO; CONDITION NOT MET
194	Ø277	Ø314	1 1ØØ 11Ø Ø11		BRN	LINTE1	ZERO; CONDITION MET
195	Ø3ØØ	Ø355	1 11Ø 111 ØØ1	LSTEP	JSB	LKYUP1	WAIT FOR KEY RELEASE
196	Ø3Ø1		Ø 111 ØØØ ØØØ	LRUN2	PINC		INCREMENT PRGM.COUNTER AND LOAD ADDR.
197	Ø3Ø2		1 ØØØ 1ØØ ØØØ		READ		READ NEW INSTRUCTION
198	Ø3Ø3		1 Ø11 Ø1Ø 1ØØ		YS11		EXTERNAL FLAG?
199	Ø3Ø4	Ø3Ø7	1 1ØØ Ø11 111		BRN	LOKY1	NO, O.K.
2ØØ	Ø3Ø5		1 ØØØ 1Ø1 11Ø	LERRA	W,BXC		YES; TROUBLE; STORE DATA
2Ø1	Ø3Ø6	Ø273	1 Ø11 1Ø1 111		BRN	LTRB1	
2Ø2	Ø3Ø7		1 Ø11 1Ø1 11Ø	LOKY1	W,ZTA		PREPARE A-REG TO ACCEPT DIGIT
2Ø3	Ø31Ø		Ø ØØ1 Ø1Ø ØØØ		EERA		ACCEPT NEW INSTRUCTION
2Ø4	Ø311	Ø263	1 Ø11 ØØ1 111		BRN	LSTOP	
2Ø5	Ø312		Ø ØØ1 111 11Ø	LIFP1	S,CM1		CHECK SIGN OF MANTISSA
2Ø6	Ø313	Ø2ØØ	1 ØØØ ØØØ Ø11		BRN	LINTE2	NEGATIVE; CONDITION NOT MET
2Ø7	Ø314		1 ØØ1 ØØØ 1ØØ	LINTE1	SS9		POSITIVE; CONDITION MET
2Ø8	Ø315	Ø2ØØ	1 ØØØ ØØØ Ø11		BRN	LINTE2	
2Ø9	Ø316		Ø 11Ø 111 11Ø	LIFM1	S,ZMC		CHECK SIGN OF MANTISSA
21Ø	Ø317	Ø2ØØ	1 ØØØ ØØØ Ø11		BRN	LINTE2	POSITIVE; CONDITION NOT MET
211	Ø32Ø	Ø314	1 1ØØ 11Ø Ø11		BRN	LINTE1	NEGATIVE; CONDITION MET
212	Ø321		1 ØØ1 ØØØ 1ØØ	LSLE1	SS9		SET LAST ENTRY FLAG ROUTINE BEGINNING
213	Ø322	Ø2Ø1	1 ØØØ ØØØ 111	LYLE1	BRN	LYLE2	INTERROGATE LAST ENTRY FLAG ROUT. BEGINNING
214	Ø323	Ø163	Ø 111 Ø1Ø ØØ1	LDSPR	JSB	LPRS1	DATA STORAGE DATA ADJ.ROUTINE
215	Ø324		Ø 1ØØ 1ØØ ØØØ		RMGRC		RETURN TO DATA STORAGE SOFTVARE
216	Ø325		Ø 1ØØ 1ØØ ØØØ	LLIST	RMGRC		LIST DATA
217	Ø326		Ø 1ØØ 1ØØ ØØØ	LCLRR	RMGRC		CLEAR DATA
218	Ø327		Ø 1ØØ 1ØØ ØØØ	LDSIN	RMGRC		D.S.INCREMENT
219	Ø33Ø		Ø 1ØØ 1ØØ ØØØ	LDSDE	RMGRC		D.S.DECREMENT
22Ø	Ø331		Ø 1ØØ 1ØØ ØØØ	LSTOR	RMGRC		STORE DIRECT BEGINNING
221	Ø332		Ø 1ØØ 1ØØ ØØØ	LSTOA	RMGRC		ACCUMULATE + BEGINNING
222	Ø333		Ø 1ØØ 1ØØ ØØØ	LSTOM	RMGRC		ACCUMULATE – BEGINNING
223	Ø334		Ø 1ØØ 1ØØ ØØØ	LSTOT	RMGRC		ACCUMULATE x BEGINNING
224	Ø335		Ø 1ØØ 1ØØ ØØØ	LSTOD	RMGRC		ACCUMULATE / BEGINNING
225	Ø336		Ø 1ØØ 1ØØ ØØØ	LRECL	RMGRC		RECALL DIRECT BEGINNING
226	Ø337		Ø 1ØØ 1ØØ ØØØ	LEXCH	RMGRC		EXCHANGE DIRECT
227	Ø34Ø		Ø 1ØØ 1ØØ ØØØ	LDECT	RMGRC		DECREMENT AND TEST IF ZERO
228	Ø341	Ø3Ø5	1 1ØØ Ø1Ø 111		BRN	LERRA	
229	Ø342		Ø ØØ1 ØØØ 1ØØ	LRTRN	SS1		RETURN ROUTINE BEGINNING
23Ø	Ø343	Ø23Ø	1 ØØ1 1ØØ Ø11		BRN	LJMPS	
231	Ø344	Ø167	Ø 111 1ØØ ØØ1	LSUB2	JSB	LPRT7	PREPARE SYMBOL – AND PRINT
232	Ø345	Ø157	Ø 11Ø 111 111		BRN	LCOD2	GOTO LOAD SECOND OP. CODE DIGIT

MAIN UNIT LISTING (ROM A—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	Ø346		1 ØØ1 ØØØ 1ØØ	LSETF	SS9		SET PROGRAM FLAG ≠ BEGINNING,SET AUX.FLG.
234	Ø347		Ø Ø1Ø 1Ø1 ØØØ	LIFFL	CXM		IF PROGRAM FLAG ≠ BEGINNING,GET INT.F.VECT.
235	Ø350		1 Ø11 ØØ1 1ØØ		PT!1		PREPARE DIGIT INTERPRETATION CODE FLAG
236	Ø351		Ø Ø11 Ø11 ØØØ		LDC3		SET IT TO THREE
237	Ø352	Ø265	1 Ø11 Ø1Ø 111		BRN	LSTP1	
238	Ø353		1 ØØ1 ØØØ 1ØØ	LGTLC	SS9		GO TO LABEL GIVEN BY C-REG
239	Ø354		1 ØØØ ØØØ 1ØØ	LGTLP	SS8		GO TO LABEL GIVEN BY PROGRAM
240	Ø355	Ø227	1 ØØ1 Ø11 111	LGTAB	BRN	LJMP1	GO TO ADDRESS GIVEN BY PROGRAM
241	Ø356		Ø ØØØ Ø1Ø 1ØØ	LKYUP1	YSØ		IS THE KEY STILL DOWN ?
242	Ø357	Ø22Ø	1 ØØ1 ØØØ Ø11		BRN	LPRTZ	NO,GO TO SUBROUTINE END
243	Ø360		Ø ØØØ 1ØØ 1ØØ		RSØ		YES, RESET FLAG
244	Ø361	Ø356	1 11Ø 111 Ø11		BRN	LKYUP1	STAY IN WAIT-LOOP
245	Ø362		1 1ØØ Ø1Ø ØØØ	LTWWS	ROM 6		X/12 ROUTINE
246	Ø363		Ø Ø1Ø Ø1Ø ØØØ	LPADV	ROM 1		PAPER ADVANCE BEGINNING POINT
247	Ø364	Ø163	Ø 111 Ø1Ø ØØ1	LEQLA	JSB	LPRS1	PREPARE DATA AND "BLANK" FOR PRINT
248	Ø365		Ø ØØØ 111 1ØØ		PLS		PREPARE SYMBOL CODE LOAD
249	Ø366		1 Ø11 Ø11 ØØØ		LDC11		R
250	Ø367		1 111 Ø11 ØØØ	LEQLG	LDC15		BLANK
251	Ø370	Ø176	Ø 111 111 1Ø1		JSB	LPRTØ	GO TO PRINT
252	Ø371		Ø 1ØØ Ø1Ø ØØØ		ROM 2		SWITCH BACK TO ROM2
253	Ø372		Ø Ø11 Ø1Ø ØØØ	LMANC	TKRA		ACCEPT KEYCODE
254	Ø373	Ø162	Ø 111 ØØ1 Ø11	LEND	BRN	LRES8	PROGRAM END ROUTINE BEGINNING
255	Ø374		1 1ØØ Ø1Ø ØØØ	LLOGS	ROM 6		LOG ROUTINE
256	Ø375		1 1ØØ Ø1Ø ØØØ	LLNS	ROM 6		LN ROUTINE
257	Ø376		1 1ØØ Ø1Ø ØØØ	LIOPC	ROM 6		I/O CALL BEGINNING
258	Ø377	Ø3Ø5	1 1ØØ Ø1Ø 111	LMEMT	BRN	LERRA	MEMORY LOCATION NOT FOUND

MAIN UNIT LISTING (ROM A—1)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 110 010 100	LNXT1	YS6		IS DEC. POINT FLAG ALREADY SET?
4	0001	0206	1 000 011 011		BRN	LNXT18	NO
5	0002		0 101 010 100		YS5		YES; IS IT DEC. POINT KEY AGAIN
6	0003	0137	0 101 111 111		BRN	LNXT2	NO
7	0004		0 011 001 110	LNXT4	W,ZTC		YES; TROUBLE; SET ZERO FOR NEW DATA
8	0005		0 000 110 100		CLS		CLEAR ALL STATUS BITS
9	0006	0043	0 010 001 111		BRN	LSUPB	GO TO SUPERVISOR
10	0007		1 111 101 010	LNXT10	X,AP1A		INCREMENT EXPONENT BY ONE
11	0010		0 000 000 000		NOP		TO ELIMINATE POSSIBLE CARRY
12	0011	0217	1 000 111 111		BRN	LNXT9	CONTINUE IN INCREMENT LOOP
13	0012		0 011 010 010	LRESA	WP,ZTC		RESET REST OF THE C-REG
14	0013		0 111 110 000		CTT		STORE NEW INT. FLG VECTOR IN T-REG TEMP
15	0014	0254	1 010 110 101		JSB	LPAPA	CALL PAPER ADVANCE SUBROUTINE
16	0015	0020	0 001 000 011		BRN	LRES8	GO TO RESET INTERNAL DATA STORAGE REGS.
17	0016	0254	1 010 110 101	LPAD1	JSB	LPAPA	CALL PAPER ADVANCE SUBROUTINE
18	0017	0106	0 100 011 011		BRN	LDIGF	GO TO SUPERVISOR
19	0020		0 100 010 000	LRES8	ROM 2		TO COMMON ADRESS PART GENERATOR
20	0021		1 001 110 000	LRES3	ATDS		CALL INT. D.S.R.
21	0022		0 010 101 000		CXM		STORE ADDRESS TEMPORARILY IN M-REG,ZERO IN C
22	0023		1 011 110 000		DTDS		CLEAR GIVEN DSR
23	0024		0 010 101 000		CXM		RESTORE ADDRESS IN C-REG,ZERO IN M-REG
24	0025		0 101 100 010		P,CM1C		DECREMENT CURRENT ADDR. AND CECK IF CARRY
25	0026	0021	0 001 000 111		BRN	LRES3	NO, CONTINUE IN DSR RESET LOOP
26	0027		1 110 010 000		ROM 7		YES, FINISHED, SWITCH TO ROM 7
27	0030	0336	1 101 111 011	LFPRA	BRN	LFPR1	
28	0031	0244	1 010 010 011	LRESD	BRN	LRESS	CONTINUE
29	0032		0 101 000 000	LRESP	IS2		PREPARE P-REG RESET
30	0033		1 000 000 000		TTP		ZEROS INTO P-REG
31	0034		1 001 001 000		PDEC		ONE'S INTO P-REG AND I/O-REG (LEFT PART)
32	0035		0 101 101 000		SRI		ONE'S INTO I/O-REG DATA PART
33	0036		1 001 000 000		IS1		END
34	0037		0 000 110 100	LRSHR	CLS		CLEAR ALL STATUS BITS
35	0040		0 011 000 100		SS3		SET OP-FLAG
36	0041	0043	0 010 001 111		BRN	LSUPB	GO TO SET FIRST ENTRY DIGIT FLAG
37	0042		0 000 110 100	LSUPA	CLS		CLEAR ALL STATUS BITS
38	0043		0 111 000 100	LSUPB	SS7		SET FIRST ENTRY DIGIT FLAG
39	0044	0106	0 100 011 011		BRN	LDIGF	
40	0045		1 001 100 100	LFIN2	RS9		RESET FLAG IF SET
41	0046		1 000 100 100		RS8		RESET FLAG IF SET
42	0047		0 110 100 100		RS6		RESET FLAG IF SET
43	0050		0 101 100 100		RS5		RESET FLAG IF SET
44	0051		0 100 100 100		RS4		RESET FLAG IF SET
45	0052		0 001 100 100		RS1		RESET FLAG IF SET
46	0053	0043	0 010 001 111		BRN	LSUPB	GO TO SUPERVISOR
47	0054		0 101 100 010	LDGT2	P,CM1C		IS IT "ONE"
48	0055	0251	1 010 100 111		BRN	LDGT3	NO; CONTINUE IN THE TEST
49	0056		0 000 011 000		LDC0		YES; RESET DIGIT INTERPRET CODE TO ZERO
50	0057		0 000 001 100		PT0		PREPARE DIGIT ALIGNMENT SHIFT COUNTER
51	0060	0157	0 111 000 001		JSB	LSHFT	
52	0061		0 101 010 100		YS5		WAS DEC. POINT KEY DOWN?

MAIN UNIT LISTING (ROM A—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0313	1 100 101 111		BRN	LRND5	NO, IT WAS DIGIT KEY
54	0063		1 001 011 000		LDC9		YES, SET SCI. NOT. FORMAT DPL. FLAG
55	0064	0322	1 101 001 011		BRN	LRND6	
56	0065		0 111 111 010	LCOU1	XS,CP1C		INCREMENT COUNTER
57	0066		0 111 110 000		CTT		RESTORE COUNT TO T REG
58	0067		0 000 110 000		RETURN		
59	0070		0 010 101 000	LRNDA	CXM		FETCH INT. FLG. VECTOR FOR UPDATE
60	0071	0202	1 000 001 011		BRN	LRND3	CONTINUE IN ROUND 0 ROUTINE
61	0072		1 011 001 100	LDGTA	PT11		PREPARE DIGIT INTERPRETATION CODE FOR TEST
62	0073		0 101 100 010		P,CM1C		IS IT ZERO?
63	0074	0054	0 010 110 011		BRN	LDGT2	NO, CONTINUE IN THE TEST
64	0075		0 010 001 110		W,BTC		YES; IT IS DATA ENTRY; RESTORE DATA IN C-REG
65	0076		0 011 100 100		RS3		RESET OP. FLAG
66	0077		0 010 100 100	LFSTT	RS2		RESET LEFT PAR. FLAG
67	0100	0107	0 100 011 111		BRN	LFST8	
68	0101	0235	1 001 110 111	LPRX3	BRN	LPRXB	
69	0102		1 001 101 010	LFST1	X,AM1		WAS ZERO ENTERED?
70	0103	0117	0 100 111 111		BRN	LFST2	NO; IT IS NONZERO DIGIT
71	0104	0042	0 010 001 011		BRN	LSUPA	YES; USE "ZERO" FOR NEW DATA
72	0105		0 111 000 110	LDIGZ	M,APCC		BUILD "NEW" DATA IN C REG
73	0106		0 000 010 000	LDIGF	ROM 0		GO TO LSUP1
74	0107		0 111 010 100	LFST8	YS7		IS THIS FIRST ENTRY DIGIT?
75	0110	0000	0 000 000 011		BRN	LNXT1	NO; GO TO "NEXT DIGIT" ROUTINE
76	0111		0 011 001 110		W,ZTC		YES; SET "ZERO" FOR "OLD" DATA
77	0112		0 111 110 000		CTT		STORE ZERO AS DISPLAY COUNT
78	0113	0174	0 111 110 011		BRN	LFST9	
79	0114		0 000 110 100	LFST7	CLS		YES; CLEAR ALL STATUS BITS; ZERO= NEW DATA
80	0115		0 110 000 100		SS6		SET DEC. POINT FLAG
81	0116	0106	0 100 011 011		BRN	LDIGF	GOTO RETURN TO SUPERVISOR
82	0117		0 111 100 100	LFST2	RS7		RESET FIRST ENTRY DIGIT FLAG
83	0120		0 001 001 100		PT1		PREPARE LEFT ALIGNMENT SHIFT COUNTER
84	0121	0157	0 111 000 001		JSB	LSHFT	
85	0122		0 110 100 000	LFST31	YADP		IS AUTO D.P. FLAG SET?
86	0123		1 011 010 100		YS11		IS IT?
87	0124	0131	0 101 100 111		BRN	LFST5	NO; IT IS NORMAL ENTRY
88	0125		1 011 100 100		RS11		YES; RESET EXTERNAL FLAG
89	0126		1 010 101 000		MTC		PREPARE DPL-FLAG FOR CHECK
90	0127		0 101 100 010	LFST4	P,CM1C		TEST DECREMENT
91	0130	0177	0 111 111 111		BRN	LFST6	NON ZERO (NO CARRY); GO TO EXP. DECREMENT
92	0131		1 010 101 000	LFST5	MTC		CARRY, DECREMENTING IS FINISHED
93	0132		1 010 001 100		PT10		GENERATE NUMBER OF ADJUSTMENT SHIFTS
94	0133		1 001 011 000		LDC9		NEEDED FOR NEXT ENTRY DIGIT
95	0134		0 010 101 000		CXM		STORE NEW INT. FLAG VECTOR
96	0135		1 110 101 110		W,AXC		SEND NEW DATA TO C-REGISTER
97	0136	0106	0 100 011 011		BRN	LDIGF	GO TO RETURN TO SUPERVISOR
98	0137		0 110 100 110	LNXT2	M,ZMC		CHECK IF MANTISSA IS ZERO
99	0140	0147	0 110 011 111		BRN	LNXT3	YES
100	0141		1 000 101 110	LNXT13	W,BXC		NO, STORE DATA TEMPORARILY IN B-REG
101	0142		1 010 101 000		MTC		PREPARE INT. FLAG VECTOR FOR TEST
102	0143		1 010 001 100		PT10		NUMBER OF NEEDED ALIGNMENT SHIFTS
103	0144		0 001 100 010		P,CM1		IS IT ZERO?
104	0145	0165	0 111 010 111		BRN	LNXT6	NO START WITH ALIGNMENT
105	0146	0307	1 100 011 111		BRN	LNXT80	
106	0147		0 101 101 010	LNXT3	X,CM1C		DECREMENT EXPONENT BY ONE
107	0150		0 001 001 100		PT1		PREPARE UNDERFLOW CHECK
108	0151		0 110 110 010		WP,ZMC		CHECK IF UNDERFLOW
109	0152	0004	0 000 010 011		BRN	LNXT4	UNDERFLOW; GO TO "TROUBLE" EXIT
110	0153	0157	0 111 000 001		JSB	LSHFT	
111	0154		0 110 011 110		S,CTA		COPY SIGN TO A-REG
112	0155		0 110 001 010		X,CTA		COPY EXPONENT TO A-REG

MAIN UNIT LISTING (ROM A—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	0156	0310	1 100 100 101		JSB	LCOU	
114	0157	0131	0 101 100 111		BRN	LFST5	GO TO "FIRST DIGIT" OUTPUT
115	0160		0 100 001 110	LSHFT	W,SLA		ALIGNMENT SHIFT
116	0161		0 000 111 100		PLS		
117	0162		1 101 101 100		YP13		ARE SHIFTS COMPLETED
118	0163	0160	0 111 000 011		BRN	LSHFT	NO, CONTINUE
119	0164		0 000 110 000		RETURN		YES, FINISHED
120	0165		0 100 001 010	LNXT6	X,SLA		PREALIGNMENT SHIFT
121	0166		0 100 001 110	LNXT7	W,SLA		ALIGNMENT SHIFT
122	0167		0 101 100 010		P,CM1C		ALIGNMENT FINISHED?
123	0170	0166	0 111 011 011		BRN	LNXT7	NOT YET
124	0171		1 010 101 000		MTC		YES, FETCH INT. FLG. VECTOR FOR UPDATE
125	0172		0 101 100 010		P,CM1C		DECREMENT NUMBER OF ALIGNMENT SHIFTS
126	0173	0305	1 100 010 111		BRN	LNXT8	
127	0174		0 101 010 100	LFST9	YS5		IS THIS DECIMAL POINT KEY ROUTINE
128	0175	0102	0 100 001 011		BRN	LFST1	NO, IT IS DIGIT ENTRY
129	0176	0114	0 100 110 011		BRN	LFST7	YES
130	0177		1 101 101 010	LFST6	X,AM1A		DECREMENT DATA EXPONENT BY ONE
131	0200		0 000 000 000		NOP		TO ELIMINATE POSSIBLE CARRY
132	0201	0127	0 101 011 111		BRN	LFST4	CONTINUE IN DECREMENT LOOP
133	0202		1 011 001 100	LRND3	PT11		PREPARE DIGIT INTERPRETATION CODE FLAG
134	0203		0 001 011 000		LDC1		SET IT TO ONE
135	0204		0 010 101 000	LCONT3	CXM		STORE UPDATED INT. FLG. VECTOR, RESTORE DATA
136	0205	0106	0 100 011 011		BRN	LDIGF	GO TO SUPERVISOR
137	0206		0 110 100 000	LNXT18	YADP		IS AUTO D.P. FLG. SET?
138	0207		1 011 010 100		YS11		IS IT?
139	0210	0302	1 100 001 011		BRN	LNXT14	NO
140	0211		1 011 100 100		RS11		YES; RESET EXTERNAL FLAG
141	0212		0 101 010 100		YS5		IS THIS DEC. POINT KEY ROUTINE?
142	0213	0223	1 001 001 111		BRN	LNXT11	NO; IT IS A DIGIT
143	0214		1 110 101 110		W,AXC		YES, STORE CURRENT DATA IN A-REG
144	0215		1 010 101 000		MTC		PREPARE INT. FLG. VECTOR FOR TEST
145	0216		1 101 001 100		PT13		PREPARE DPL FLG FOR TEST
146	0217		0 101 100 010	LNXT9	P,CM1C		DECREMENT AND TEST IF CARRY OCCURRED
147	0220	0007	0 000 011 111		BRN	LNXT10	NO CARRY, CONTINUE IN EXP. INCREMENT
148	0221		1 110 101 110		W,AXC		CARRY; FINISHED; RESTORE DATA IN C-REG
149	0222	0114	0 100 110 011		BRN	LFST7	GO TO D.P. FLAG OUTPUT
150	0223		0 111 101 010	LNXT11	X,CP1C		INCREMENT CURRENT EXPONENT BY ONE
151	0224		0 101 111 010		XS,CM1C		DECREMENT CODE FOR SIGN OF EXP.
152	0225		0 110 111 010		XS,ZMC		IS IT ZERO NOW?
153	0226	0004	0 000 010 011		BRN	LNXT4	OVERFLOW; GO TO "TROUBLE" EXIT
154	0227		0 111 111 010		XS,CP1C		NO, RESTORE ORIG VALUE
155	0230		0 000 000 000		NOP		TO ELIMINATE POSSIBLE CARRY
156	0231	0141	0 110 000 111		BRN	LNXT13	NO OVERFLOW DETECTED
157	0232		0 010 011 000	LEQLR	LDC2		
158	0233		0 000 010 000	LEQLZ	ROM 0		
159	0234		0 000 110 100		CLS		CLEAR STATUS BITS
160	0235		0 010 001 110	LPRXB	W,BTC		RESTORE ORIGINAL DATA IN C-REGISTER
161	0236		0 110 100 100		RS6		RESET D.P. FLAG
162	0237	0241	1 010 000 111		BRN	LPRXL	GO TO TEST
163	0240	0235	1 001 110 111	LEQLF	BRN	LPRXB	GO TO COMMON PATH WITH PRINT x ROUTINE
164	0241		0 010 010 100	LPRXL	YS2		IS L.PAR FLAG SET?
165	0242	0043	0 010 001 111		BRN	LSUPB	GO TO SET FIRST ENTRY DIGIT FLAG AND SUPERVISOR
166	0243		0 110 010 000		ROM 3		RETURN TO LEFT PARENTHESIS ROUTINE
167	0244		0 011 001 110	LRESS	W,ZTC		CLEAR C-REG
168	0245		0 111 110 000		CTT		CLEAR T-REG
169	0246		0 100 010 100		YS4		IS THIS RESET ROUTINE?
170	0247	0032	0 001 101 011		BRN	LRESP	NO, GO TO PRGM. COUNTER RESET
171	0250	0037	0 001 111 111		BRN	LRSHR	YES, SKIP PRGM. COUNTER RESET
172	0251		0 101 100 010	LDGT3	P,CM1C		IS IT TWO?

MAIN UNIT LISTING (ROM A—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252	0265	1 011 010 111		BRN	LDGT4	NO; CONTINUE
174	0253		0 000 011 000		LDC0		YES; IT IS I/O CALL; RESET D.I. CODE TO ZERO
175	0254	0365	1 111 010 111		BRN	LDGT30	GO TO I/O CHANNEL SELECT AND CALL ROUTINE
176	0255		0 101 110 000	LPAPA	PRE		PRINTER MECH. ENABLE
177	0256		1 011 010 100		YS11		FLAG?
178	0257	0255	1 010 110 111		BRN	LPAPA	NO, WAIT
179	0260	0277	1 011 111 111		BRN	LPAD2	YES; CONTINUE
180	0261		0 011 010 100	LCENA	YS3		INTERROGATE OP. FLAG
181	0262	0004	0 000 010 011		BRN	LNXT4	NOT SET,GO TO CLEAR C-REG
182	0263	0043	0 010 001 111		BRN	LSUPB	SET,DO NOTHING
183	0264		1 000 010 000	LFLG1	ROM 4		
184	0265		0 101 100 010	LDGT4	P,CM1C		IS IT THREE?
185	0266	0361	1 111 000 111		BRN	LDGT5	NO; CONTINUE
186	0267	0264	1 011 010 011		BRN	LFLG1	YES, GO TO ACCEPT PRGM.FLAG NUMBER
187	0270		0 001 000 100	LPRINA	SS1		SET SHIFT KEY ROUTINES AUX.FLG
188	0271		1 111 011 000		LDC15		
189	0272	0336	1 101 111 011		BRN	LFPR1	
190	0273		0 010 101 000	LCONTA	CXM		PREPARE INT.FLG.VECTOR FOR UPDATE
191	0274		0 001 011 000		LDC1		SET PRGM.RUN MODE FLAG
192	0275	0376	1 111 111 011		BRN	LCONT39	
193	0276	0045	0 010 010 111	LFINA	BRN	LFIN2	GO TO RESET FLAGS
194	0277		1 011 100 100	LPAD2	RS11		RESET EXT FLAG
195	0300		0 011 110 000		ADV		PAPER ADVANCE
196	0301		0 000 110 000		RETURN		SUBROUTINE END
197	0302		0 101 010 100	LNXT14	YS5		IS THIS DEC. POINT KEY ROUTINE?
198	0303	0223	1 001 001 111		BRN	LNXT11	NO, ACCEPT DIGIT
199	0304	0114	0 100 110 011		BRN	LFST7	YES, GO TO D.P. FLAG OUTPUT
200	0305		0 010 101 000	LNXT8	CXM		STORE NEW INTERNAL FLAG VECTOR
201	0306	0310	1 100 100 101		JSB	LCOU	
202	0307		0 010 001 110	LNXT80	W,BTC		RESTORE OLD DATA IN C REG
203	0310	0105	0 100 010 111		BRN	LDIGZ	
204	0311		0 111 111 000	LCOU	TTC		GET DISPLAY COUNT
205	0312	0065	0 011 010 111		BRN	LCOU1	
206	0313		0 011 011 000	LRND5	LDC3		LOAD THREE FOR TEST
207	0314		0 000 111 100		PLS		RESTORE POINTER
208	0315		1 111 000 010		P,APCA		IS DIGIT GREATER THAN SIX
209	0316	0321	1 101 000 111		BRN	LDPT3	NO, RESTORE IT
210	0317		0 110 011 000		LDC6		YES, SET DIGIT TO SIX
211	0320	0322	1 101 001 011		BRN	LRND6	
212	0321		0 101 000 010	LDPT3	P,AMCC		
213	0322		0 101 100 100	LRND6	RS5		RESET D.P. KEY FLAG IF SET
214	0323		0 010 101 000		CXM		STORE NEW INT. FLG. VECTOR IN M-REG
215	0324		0 010 001 110		W,BTC		RESTORE DATA IN C-REG
216	0325	0106	0 100 011 011		BRN	LDIGF	GO TO LSUP1
217	0326		0 100 101 110	LCLAA	W,ATB		COPY LABEL BACK INTO B-REG
218	0327		1 011 010 110		MS,SRA		PREALIGNMENT
219	0330		1 011 010 110		MS,SRA		PREALIGNMENT
220	0331		0 100 010 110	LCLA1	MS,SLA		ALIGNMENT SHIFT
221	0332		1 101 101 010		X,AM1A		DECREMENT EXPONENT
222	0333	0331	1 101 100 111		BRN	LCLA1	NO CARRY; ALIGNMENT CONTINUE
223	0334		1 110 101 110		W,AXC		GET ADJUDTED LABEL INTO C-REG
224	0335		1 100 010 000		ROM 6		CARRY; ALIGNMENT FINISHED
225	0336		0 000 001 100	LFPR1	PT0		PREPARE SECTOR COUNTER
226	0337		0 101 110 000		PRE		PRINTER ENABLE
227	0340		1 011 010 100	LFPR2	YS11		FLAG?
228	0341	0340	1 110 000 011		BRN	LFPR2	NO, WAIT
229	0342		1 011 100 100		RS11		YES; RESET EXTERNAL FLAG
230	0343		1 111 110 000		CCS		RIGHT PART OF THE PRINTER MASK
231	0344		1 101 110 000		TCS		LEFT PART OF THE PRINTER MASK
232	0345		0 000 111 100		PLS		INCREMENT SECTOR COUNTER

MAIN UNIT LISTING (ROM A—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		1 101 101 100		YP13		LAST SECTOR?
234	0347	0340	1 110 000 011		BRN	LFPR2	NO
235	0350		1 011 010 100	LFPR3	YS11		YES, FLAG?
236	0351	0350	1 110 100 011		BRN	LFPR3	NO, WAIT
237	0352		1 011 100 100		RS11		YES, RESET EXTERNAL FLAG
238	0353		0 011 110 000		ADV		PAPER ADVANCE
239	0354		0 001 010 100		YS1		IS SHIFT KEY ROUT.AUX.FLG. SET?
240	0355	0366	1 111 011 011		BRN	LFPR4	
241	0356		0 001 100 100		RS1		RESET SHIFT KEY ROUTINES AUX.FLG.
242	0357		0 010 001 110		W,BTC		
243	0360		1 100 010 000		ROM 6		SWITCH TO ROM 6
244	0361		1 100 010 000	LDGT5	ROM 6		
245	0362	0367	1 111 011 111	LEQLD	BRN	LEQLK	
246	0363		0 000 000 000		DUMMY		
247	0364	0016	0 000 111 011	LPADA	BRN	LPAD1	GO TO PAPER ADVANCE ROUTINE
248	0365		0 100 010 000	LDGT30	ROM 2		GO TO I/O CHANNEL SELECT AND CALL ROUTINE
249	0366		1 110 010 000	LFPR4	ROM 7		
250	0367		1 010 000 100	LEQLK	SS10		SET SECOND PRINT FLAG
251	0370		1 000 010 100		YS8		IS RIGHT PARENTHESIS FLAG SET?
252	0371	0232	1 001 101 011		BRN	LEQLR	
253	0372		0 000 111 100		PLS		YES,PREPARE SYMBOL LOAD
254	0373		0 011 011 000		LDC3		SYMBOL FOR RIGHT PAR.)
255	0374		1 111 011 000		LDC15		BLANK
256	0375	0233	1 001 101 111		BRN	LEQLZ	GO TO COMMON PART
257	0376		0 000 101 000	LCONT39	DSTO		SWITCH THE DISPLAY ON
258	0377	0204	1 000 010 011		BRN	LCONT3	

MAIN UNIT LISTING (ROM A—2)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 000 010 000		ROM 0		GO TO ROM 0 AFTER ROM GROUP SWITCH
4	0001		1 100 001 100	LIDS1	PT12		PREPARE SPEC. ADDR. CODE LOAD
5	0002		0 011 001 110	LIDS2	W,ZTC		ZERO TO C-REG
6	0003		0 111 101 110		W,CP1C		SET EXPONENT TO "ONE"
7	0004		1 111 011 000		LDC15		LOAD SPEC. ADDR. FOR INT. D.S.
8	0005		0 000 110 000		RETURN		END OF SUBROUTINE
9	0006		1 010 101 000	LRIP3	MTC		GET INT. FLAG VECTOR FOR TEST
10	0007		1 001 001 100		PT9		PREPARE LEVEL INDICATOR FOR TEST
11	0010		0 101 100 010		P,CM1C		CHECK IF LEVEL INDICATOR IS ZERO
12	0011	0241	1 010 000 111		BRN	LRIPA	IT IS NOT ZERO, CONTINUE
13	0012		1 110 010 000	LRIP4	ROM 7		SWITCH TO ROM7
14	0013		0 000 111 100	LRES69	PLS		RESTORE PROPER POINTER VALUE
15	0014	0020	0 001 000 011		BRN	LRES6	PREPARE RETURN TO RESET ROUTINE
16	0015		1 000 000 100	LRIP5	SS8		SET RIGHT PARENTHESIS FLAG
17	0016		0 010 001 110		W,BTC		RESTORE DATA IN C-REG
18	0017		0 000 010 000		ROM 0		GO TO COMMON PATH WITH "EQUAL" ROUTINE
19	0020		0 010 010 000	LRES6	ROM 1		GO TO RESET ROUTINE
20	0021	0047	0 010 100 001	LRESE	JSB	LIDS10	GO TO ADDRESS GENERATING SUBROUTINE
21	0022	0354	1 110 110 011		BRN	LRES9	
22	0023		0 110 100 000	LDSP50	YADP		IS AUTO DEC. PT. SET?
23	0024		1 011 010 100		YS11		
24	0025	0027	0 001 011 111		BRN	LDSP5	NO, GO TO NORMAL PATH
25	0026		0 010 001 100		PT2		YES, SET POINTER FOR ALL OF MANTISSA
26	0027		0 010 101 100	LDSP5	YP2		IS ALL OF MANTISSA TO BE DISPLAYED
27	0030	0315	1 100 110 111		BRN	LDSP51	
28	0031	0126	0 101 011 011		BRN	LDSP6	YES, POINTER IS SET
29	0032	0006	0 000 011 011		BRN	LRIP3	
30	0033		1 100 100 000	LMAN6	YFKB		INTERROGATE F. BL. KBD. AND SET BUSY LIGHT
31	0034		1 011 010 100		YS11		IS IT?
32	0035	0167	0 111 011 111		BRN	LMAN7	NO
33	0036		1 100 010 000		ROM 6		GO TO CALL F.BLK.SOFTWARE
34	0037		1 011 010 100	LCLO3	YS11		CHECK IF SINGLE OPERAND OP. FLAG IS SET
35	0040	0267	1 011 011 111		BRN	LCLOA	
36	0041		0 010 010 000	LMTX3	ROM 1		GO TO LSUPA
37	0042		0 010 010 000	LCLO6	ROM 1		GO TO SUPERVISOR
38	0043		1 010 100 100	LMAN9	RS10		RESET SHIFT KEY FLAG
39	0044		1 100 010 000		ROM 6		GO TO ROM 6
40	0045		0 011 011 000	LBAK1	LDC3		MASK,PART 2
41	0046		1 111 011 000		LDC15		MASK,PART 3
42	0047	0356	1 110 111 011		BRN	LBAK2	
43	0050		1 110 101 000	LIDS10	CLR		CLEAR ALL REGISTERS
44	0051	0001	0 000 000 111		BRN	LIDS1	
45	0052		1 011 101 110	LALFA	W,ZTA		PREPARE C-REG FOR MESSAGE
46	0053		0 111 000 000	LALF1	PINC		INCREMENT PRGM COUNTER
47	0054		0 000 011 100		PRS		DECREMENT COUNTER
48	0055		0 110 101 100		YP6		FINISHED?
49	0056	0053	0 010 101 111		BRN	LALF1	NO; CONTINUE
50	0057		1 100 010 000		ROM 6		YES; GO TO ROM 6 TO CONTINUE
51	0060	0263	1 011 001 111	LCLOP	BRN	LCLO8	
52	0061		0 000 000 000		DUMMY		

MAIN UNIT LISTING (ROM A—2) Continued

```
Line  Curr  Bran  Operation Code
 #    Addr  Addr  Bit Pattern

 53  0062  0000  0 000 000 101  LXTMA  JSB     LIDS1   JUMP TO INT. D.S. ADDRESS GENERATOR
 54  0063        1 001 011 000        LDC9            CONSTANT REGISTER ADDRESS
 55  0064        1 001 110 000        ATDS            D.S. CALL
 56  0065        1 001 010 100        YS9             CHECK IF "STORE" ROUTINE FLAG IS SET
 57  0066  0364  1 111 010 011        BRN     LMTX2   NOT SET GO TO "RECALL" ROUTINE
 58  0067        0 010 001 110        W,BTC           SET, RESTORE DATA IN C-REG
 59  0070        1 011 110 000        DTDS            STORE DATA IN D.S.
 60  0071  0041  0 010 000 111        BRN     LMTX3   GO TO ROUTINE EXIT
 61  0072        1 000 101 000  LDSP1  DSOF            SWITCH THE DISPLAY OFF
 62  0073        0 110 001 110        W,CTA             COPY DATA INTO A-REG FOR ADJUSTMENT
 63  0074        0 100 101 110        W,ATB             STORE ORIG. DATA TEMPORARILY IN
                                                                                  B-REG.
 64  0075        1 011 000 100        SS11              SET "DISPLAY ROUTINE" FLAG
 65  0076        1 010 101 000        MTC               GET INTERNAL FLAG VECTOR FOR TEST
 66  0077        1 110 010 000        ROM 7             GO TO COMMON PART WITH PRINT
                                                                              SUBROUTINE
 67  0100        0 110 111 010  LOFL2  XS,ZMC           CHECK SIGN OF THE RESULT EXPONENT
 68  0101  0204  1 000 010 011        BRN     LCLO1   ZERO; POSITIVE, RESULT IS O.K.
 69  0102        0 010 101 010        X,ZMCC          CARRY; CONTINUE TEST; EXPONENT
                                                                          COMPLEMENTED
 70  0103        0 101 111 010        XS,CM1C         SIGN CODE DECREMENT
 71  0104  0251  1 010 100 111        BRN     LOFL3   NO CARRY, TROUBLE
 72  0105  0110  0 100 100 011        BRN     LCLOY   CARRY,NEG.RESULTS,O.K.
 73  0106        0 000 010 000  LSSUP  ROM 0           GO TO SUPERVISOR
 74  0107  0106  0 100 011 011        BRN     LSSUP   ACCEPT KEYCODE FROM OUTSIDE
 75  0110        1 110 101 110  LCLOY  W,AXC           RESTORE DATA IN C-REG
 76  0111  0204  1 000 010 011        BRN     LCLO1   GO TO CLOSING ROUTINE
 77  0112  0213  1 000 101 111  LDSPZ  BRN     LDSPA   GO TO GENERATE DISPLAY MASK
 78  0113        1 011 000 100  LIOS1  SS11            SET I/O CALL FLAG
 79  0114        0 000 010 000        ROM 0           GO TO I/O CALL SEQUENCE
 80  0115        1 100 001 100  LDSPP  PT12            PREPARE SEARCH FOR LEADING ZEROS
 81  0116        1 001 100 010  LDSP3  P,AM1             ZERO?
 82  0117  0177  0 111 111 111        BRN     LDSP4     NO; END OF THE SEARCH
 83  0120        0 001 100 010        P,CM1             YES; DEC. POINT?
 84  0121  0177  0 111 111 111        BRN     LDSP4     YES; END OF THE SEARCH
 85  0122        1 001 011 000        LDC9              NO; LOAD CODE FOR "BLANK"
 86  0123  0116  0 100 111 011        BRN     LDSP3     SEARCH ROUTINE
 87  0124        0 101 111 010  LDSP52 XS,CM1C         DECREMENT DISPLAY COUNTER
 88  0125  0027  0 001 011 111        BRN     LDSP5   CONTINUE
 89  0126        1 100 101 000  LDSP6  DNR             RESTORE MASK TO C REG
 90  0127        0 101 110 010        WP,CM1C           INSERT CODES FOR "BLANK"
 91  0130        1 000 101 110        W,BXC             ORIG. DATA BACK TO C-REG; MASK TO
                                                                                  B-REG
 92  0131        0 110 011 110        S,CTA             COPY ORIG. SIGN TO A-REG
 93  0132        0 101 010 100        YS5               SCI. NOTATION FORMAT?
 94  0133  0144  0 110 010 011        BRN     LDSP8     NO
 95  0134        0 000 110 010        WP,ZTB          YES,CHANGE MASK TO DISPLAY ALL
 96  0135        0 110 001 010        X;CTA             COPY ORIG. EXPONENT TO A-REG
 97  0136        0 110 111 010        XS,ZMC            CHECK IF EXPONENT IS NEGATIVE
 98  0137  0143  0 110 001 111  LWAIT  BRN     LDSP9     NO, IT IS POSITIVE, O.K.
 99  0140        0 001 001 100        PT1               YES, PREPARE COMPLEMENT
100  0141        0 010 110 010        WP,ZMCC         COMPLEMENT OF THE EXPONENT
101  0142        1 110 110 010        WP,AXC            ORIG. EXPONENT BACK TO C-REG; COMPL.
                                                                              TO A-REG
102  0143        0 101 100 100  LDSP9  RS5               RESET SCI. NOT. FORMAT FLAG
103  0144        0 000 101 000  LDSP8  DSTO              DISPLAY ON
104  0145        1 011 100 100  LDSP7  RS11            RESET DISPLAY FLAG
105  0146        0 000 110 000        RETURN          END OF THE DISPLAY ROUTINE
106  0147  0321  1 101 000 111  LPERD  BRN     LPERO
107  0150  0071  0 011 101 001  LMANB  JSB     LDSP1   JUMP TO DISPLAY SUBROUTINE
108  0151        0 100 100 000        RMGRC
109  0152        0 000 100 100  LMAN2  RS0
110  0153        0 000 010 100        YS0             IS THE "OLD" KEY STILL DOWN?
111  0154  0156  0 110 111 011        BRN     LMAN3   NO, CONTINUE
112  0155  0152  0 110 101 011        BRN     LMAN2   YES, WAIT
```

MAIN UNIT LISTING (ROM A—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 101 000 000	LMAN3	IS2		
114	0157		0 100 100 000		RBL		RESET BUSY LIGHT
115	0160		1 001 000 000		IS1		
116	0161		0 000 010 100	LMAN4	YS0		NEW KEY DOWN?
117	0162	0161	0 111 000 111		BRN	LMAN4	NO,WAIT
118	0163	0136	0 101 111 101		JSB	LWAIT	YES,GO TO DO NOTHING
119	0164		1 100 001 100		PT12		SET PROPER POINTER
120	0165	0033	0 001 101 111		BRN	LMAN6	YES
121	0166		0 000 000 000		DUMMY		
122	0167		1 010 010 100	LMAN7	YS10		WAS SHIFT KEY SET ?
123	0170	0370	1 111 100 011		BRN	LMAN8	NO,GO TO ROM 0
124	0171	0043	0 010 001 111		BRN	LMAN9	YES
125	0172		1 010 010 100	LCLO2	YS10		CHECK IF SECOND PRINT FLAG IS SET
126	0173	0037	0 001 111 111		BRN	LCLO3	NO
127	0174		0 100 010 100	LCLO7	YS4		CHECK IF RIGHT PARENTHESIS FLAG IS SET
128	0175	0312	1 100 101 011		BRN	LEQLT	
129	0176	0221	1 001 000 111		BRN	LRPAR	YES,CONTINUE IN RIGHT PARENTHESIS ROUTINE
130	0177		0 100 101 000	LDSP4	CTS		SAVE MASK IN D REG
131	0200		1 011 100 100		RS11		
132	0201		0 111 111 000	LDSP40	TTC		GET DISPLAY COUNTER IN C REG
133	0202	0023	0 001 001 111		BRN	LDSP50	
134	0203		0 000 010 000	LPER3	ROM 0		
135	0204		0 001 010 100	LCLO1	YS1		CHEKC IF ODD-AUX-FLG IS SET
136	0205	0263	1 011 001 111		BRN	LCLO8	NO; NORMAL PATH
137	0206		0 101 111 110		S,CM1C		YES,SET NEGATIVE SIGN OF THE RESULT
138	0207		0 001 100 100		RS1		YES, RESET FLAG
139	0210	0263	1 011 001 111		BRN	LCLO8	
140	0211		1 100 101 000	LLDDZ	DNR		RESTORE PREVIOUS RESULT IN C-REG
141	0212	0172	0 111 101 011		BRN	LCLO2	
142	0213		0 011 001 110	LDSPA	W,ZTC		START TO GENERATE DISPLAY MASK
143	0214		0 000 111 100		PLS		ADJUST DEC. POINT LOCATION POINTER
144	0215		0 010 011 000		LDC2		SET DISPLAY MASK CODE FOR DEC. POINT
145	0216	0115	0 100 110 111		BRN	LDSPP	
146	0217		0 000 000 000		DUMMY		
147	0220		1 110 010 000	LER20	ROM 7		
148	0221		0 000 010 000	LRPAR	ROM 0		
149	0222		0 000 000 000		DUMMY		
150	0223		0 000 000 000		DUMMY		
151	0224		0 000 000 000		DUMMY		
152	0225		0 001 100 110	LOFL1	M,CM1		CHECK IF MANTISSA IS ZERO
153	0226	0100	0 100 000 011		BRN	LOFL2	NO; CONTINUE TEST
154	0227	0253	1 010 101 111		BRN	LOFL4	YES; GO TO SET ZERO FOR RESULT
155	0230		0 000 010 000	LMTX20	ROM 0		
156	0231		0 100 010 100	LJAB2	YS4		ADDRESS MODIFICATION NEEDED ?
157	0232	0333	1 101 101 111		BRN	LRET3	NO
158	0233		0 101 000 000		IS2		YES
159	0234		1 000 101 000		PTT		CURRENT ADDRESS FOR MODIFICATION
160	0235		0 001 101 000		ADD		DODIFICATION BY BINARY ADDITION
161	0236		1 001 000 000		IS1		PREPARE MASK-OUT BITS 11 TO 16
162	0237		0 000 011 000		LDC0		MASK,PART 1 (POINTER WAS 3)
163	0240	0045	0 010 010 111		BRN	LBAK1	
164	0241		0 011 010 100	LRIPA	YS3		CHECK IF OP. FLG. IS SET
165	0242	0015	0 000 110 111		BRN	LRIP5	NO CARRY, O.K., IT IS NOT ZERO, CONTINUE
166	0243		1 101 001 100		PT13		YES, SET POINTER TO GENERATE PROPER MESS.
167	0244	0220	1 001 000 011		BRN	LER20	
168	0245		0 100 101 000	LEQLC	CTS		SAVE RESULT
169	0246	0000	0 000 000 101		JSB	LIDS1	GENERATE ADDRESS
170	0247		1 000 011 000		LDC8		
171	0250		1 110 010 000		ROM 7		
172	0251		0 001 111 010	LOFL3	XS,CM1		CHECK SIGN OF EXP. AGAIN

MAIN UNIT LISTING (ROM A—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252	0353	1 110 101 111		BRN	LOFL5	NO CARRY, I.E. OVERFLOW DETECTED
174	0253		0 011 001 110	LOFL4	W,ZTC		CARRY; UNDERFLOW; SET ZERO FOR RESULT
175	0254		0 000 000 000		NOP		TO ELIMINATE CARRY CAUSED BY THE HARDWARE
176	0255	0263	1 011 001 111		BRN	LCLO8	GO TO THE CLOSING PART OF THE ROUTINE
177	0256		0 000 000 000		DUMMY		
178	0257		0 000 000 000		DUMMY		
179	0260		0 000 000 000		DUMMY		
180	0261		0 110 010 000	LCLOG	ROM 3		
181	0262	0306	1 100 011 011	LADJ7	BRN	LADJ17	
182	0263		0 110 010 100	LCLO8	YS6		CHECK IF L-FLAG IS SET
183	0264	0172	0 111 101 011		BRN	LCLO2	NO
184	0265		1 011 110 000	LLDDA	DTDS		STORE RESULT IN D.S.
185	0266	0211	1 000 100 111		BRN	LLDDZ	
186	0267		0 100 101 000	LCLOA	CTS		COPY C TO D-REGISTER
187	0270		0 010 101 000	LCLO5	CXM		GET INT. FLG VECTOR AND STORE RESULT IN M-REG
188	0271		1 110 101 110		W,AXC		PREPARE INT. FLAG VECTOR FOR UPDATING
189	0272		0 011 001 100		PT3		SET POINTER TO INDICATE OP. CODE
190	0273		0 100 010 010		WP,SLA		"NEW" OP. CODE TRANSFERRED
191	0274		0 100 010 010		WP,SLA		TO OP. CODE "TO BE EXECUTED" NEXT
192	0275		1 110 101 110		W,AXC		UPDATED INT. FLG. VECTOR TO C-REG
193	0276		0 010 101 000		CXM		RESTORE RESULT IN C-REG; INT. FLG VECTOR TO M-REG.
194	0277		0 100 010 100		YS4		INTERROGATE AUX. FLAG
195	0300	0303	1 100 001 111		BRN	LCLOM	TECHNICAL BRANCH
196	0301		1 000 101 110		W,BXC		
197	0302	0261	1 011 000 111		BRN	LCLOG	
198	0303		0 000 110 100	LCLOM	CLS		CL. ALL FLAGS(FOR TWO OP. ROUTINES)
199	0304		0 011 000 100		SS3		SET OP. FLAG
200	0305	0042	0 010 001 011		BRN	LCLO6	GO TO ROUTINE EXIT
201	0306		1 010 111 010	LADJ17	XS,CPCC		CHECK IF ROUND UP IS NEEDED
202	0307	0376	1 111 111 011		BRN	LADJ18	NO
203	0310		0 111 100 110		M,CP1C		YES
204	0311	0376	1 111 111 011		BRN	LADJ18	
205	0312		0 011 010 100	LEQLT	YS3		IS OP.FLG.SET ?
206	0313	0363	1 111 001 111		BRN	LEQL3	NO
207	0314	0270	1 011 100 011		BRN	LCLO5	YES
208	0315		0 000 011 100	LDSP51	PRS		DECREMENT POINTER
209	0316		0 111 010 100		YS7		IS THIS A DATA ENTRY
210	0317	0124	0 101 010 011		BRN	LDSP52	YES, CONTINUE TEST
211	0320	0027	0 001 011 111		BRN	LDSP5	NO, CONTINUE DECREMENT
212	0321		0 110 100 000	LPERO	YADP		CHECK IF AUTO D.P. FLAG SET
213	0322		1 011 010 100		YS11		IS IT SET?
214	0323	0203	1 000 001 111		BRN	LPER3	NO, GO TO NORMAL PATH
215	0324		1 011 100 100		RS11		YES, RESET EXTERNAL FLAG
216	0325		0 111 010 100		YS7		IS IT MANUAL DATA ENTRY
217	0326	0330	1 101 100 011		BRN	LPER32	YES
218	0327	0203	1 000 001 111		BRN	LPER3	NO; GO TO NORMAL PATH
219	0330		0 110 010 100	LPER32	YS6		WAS DEC POINT SET?
220	0331	0335	1 101 110 111		BRN	LPER33	NO;
221	0332	0203	1 000 001 111		BRN	LPER3	YES; GO TO NORMAL PATH
222	0333		1 000 000 000	LRET3	TTP		NEW ADDRESS INTO PRGM.COUNTER
223	0334		1 100 010 000		ROM 6		
224	0335		0 110 001 110	LPER33	W,CTA		STORE DATA TEMP IN A-REG (FOR PROCESSING)
225	0336		1 010 101 000		MTC		GET INT.FLG.VECTOR FOR TEST
226	0337		1 101 001 100		PT13		PREPARE D.P.L. FLAG TEST
227	0340	0342	1 110 001 011		BRN	LPER35	
228	0341		1 111 101 010	LPER34	X,AP1A		INCREMENT EXPONENT
229	0342		0 101 100 010	LPER35	P,CM1C		DECREMENT D.P.L. FLAG
230	0343	0341	1 110 000 111		BRN	LPER34	NO CARRY
231	0344		1 110 101 110		W,AXC		RESTORE ADJUSTED DATA IN C-REG
232	0345	0203	1 000 001 111		BRN	LPER3	

MAIN UNIT LISTING (ROM A—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
233	0346		0 000 000 000		DUMMY	
234	0347		1 001 101 010	LDGTC	X,AM1	CHECK ENTERED DIGIT
235	0350	0113	0 100 101 111		BRN LIOS1	NO CARRY, CALL I/O
236	0351		0 000 001 100		PT0	CARRY, START ALPHA ROUTINE
237	0352	0052	0 010 101 011		BRN LALFA	
238	0353		1 110 010 000	LOFL5	ROM 7	
239	0354		1 000 011 00	LRES9	LDC8	PREPARE HIGHEST D.S. ADDR. TO BE RESET
240	0355	0013	0 000 101 111		BRN LRES69	GO TO RESTORE POINTER
241	0356		1 111 011 000	LBAK2	LDC15	MASK, PART 4
242	0357		0 101 000 000		IS2	
243	0360		0 010 101 000		AND	MASK- OUT BITS 11 THROUGH 16
244	0361		1 001 000 000		IS1	
245	0362	0333	1 101 101 111		BRN LRET3	
246	0363		0 000 010 000	LEQL3	ROM 0	SWITCH TO ROM0
247	0364		1 011 111 000	LMTX2	DSTC	READ DATA OUT OF D.S. INTO C-REG
248	0365	0230	1 001 100 011		BRN LMTX20	
249	0366		0 010 101 000	LDGTA	CXM	STORE UPDATED INT.FLG.VECTOR
250	0367	0347	1 110 011 111		BRN LDGTC	
251	0370		1 011 101 110	LMAN8	W,ZTA	PREPARE A-REG TO ACCEPT DIGIT
252	0371		0 000 010 000		ROM 0	GO TO ACCEPT KEYCODE
253	0372		0 011 001 110	LEQLB	W,ZTC	PREPARE ZERO IN C-REG
254	0373		0 100 101 000		CTS	LOAD IT INTO D-REG
255	0374		0 010 001 110		W,BTC	RESTORE DATA IN C-REG
256	0375	0245	1 010 010 111		BRN LEQLC	DO A ACC +
257	0376		1 110 010 000	LADJ18	ROM 7	
258	0377	0204	1 000 010 011		BRN LCLO1	

MAIN UNIT LISTING (ROM A—3)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
3	0000		0 000 000 000		DUMMY			
4	0001		0 010 001 110	LERR37	W,BTC			RESTORE ORIG DATA DAMAGED BY TEST
5	0002	0221	1 001 000 111		BRN	LERR30		
6	0003		0 101 100 010	LERR33	P,CM1C			DECREMENT LAST INT. DIGIT
7	0004	0271	1 011 100 111		BRN	LERR34		NO CARRY, TEST CONTINUE
8	0005		0 001 000 100		SS1			CARRY, ODD, SET ODD AUX. FLAG
9	0006	0010	0 000 100 011		BRN	LGO01		
10	0007		1 010 010 000	LGO02	ROM 5			
11	0010		0 000 111 110	LGO01	S,ZTB			
12	0011		0 010 001 110		W,BTC			
13	0012		1 100 001 100		PT12			
14	0013	0007	0 000 011 111		BRN	LGO02		
15	0014		1 001 001 110	LCLA3	W,SRC			PREPARE LABEL HEAD LOADING
16	0015		1 001 001 110		W,SRC			PREPARE LABEL HEAD LOADING
17	0016		1 101 001 100		PT13			PREPARE LABEL HEAD LOADING
18	0017		1 000 011 000		LDC8			LOAD 8
19	0020	0025	0 001 010 111		BRN	LCLA4		BRANCH (HARDWARE NOP)
20	0021		1 000 010 000	LEEQQ	ROM 4			
21	0022		1 000 010 100		YS8			IS IT EQUAL ROUTINE ?
22	0023	0062	0 011 001 011		BRN	LEQLP		YES,GO TO LEVEL TEST
23	0024	0021	0 001 000 111		BRN	LEEQQ		NO, SKIP LEVEL TEST
24	0025		0 111 011 000	LCLA4	LDC7			LOAD 7
25	0026	0372	1 111 101 011		BRN	LCLA20		
26	0027		1 000 101 110	LPARA	W,BXC			SAVE DATA TEMPORARILY IN B-REG
27	0030		1 010 101 000		MTC			GET IN. FLG. VECTOR FOR TEST
28	0031		1 001 001 100		PT9			PREPARE LEVEL INDICATOR FOR TEST
29	0032		1 110 100 010		P,AXC			LEVEL INDICATOR TO A-REG
30	0033		0 101 011 000		LDC5			LIMIT SET TO C-REG
31	0034		0 000 111 100		PLS			POINTER ADJUSTMENT
32	0035		0 001 000 010		P,AMC			COMPARISON
33	0036	0371	1 111 100 111		BRN	LSER1		NO CARRY, IT IS FIVE, GO TO ERROR ROUTINE
34	0037	0310	1 100 100 011		BRN	LPAR2		CARRY, IT IS LESS THAN FIVE, CONTINUE
35	0040	0212	1 000 101 011	LPERA	BRN	LPER2		PREPARE SYMBOL FOR PRINT
36	0041		0 010 010 000	LFIN2	ROM 1			GO TO LSUPA
37	0042		0 010 010 000	LFIN3	ROM 1			GO TO SUPERVISOR LSUPB
38	0043		1 000 101 110	LPER3	W,BXC			RESTORE RESULT IN C-REG
39	0044		0 101 101 010		X,CM1C			DECREMENT DATAEXPONENT
40	0045		0 101 101 010		X,CM1C			BY TWO
41	0046		1 000 101 110		W,BXC			GET INT. FLG VECTOR BACK TO C-REG
42	0047		1 001 001 110		W,SRC			OP. CODE ALIGNMENT SHIFT
43	0050		1 001 001 110		W,SRC			OP. CODE ALIGNMENT SHIFT
44	0051		0 110 100 100		RS6			
45	0052		0 101 110 010		WP,CM1C			IS IT ZERO ?
46	0053	0352	1 110 101 011		BRN	LPER4		NO, CONTINUE TEST
47	0054		0 010 001 110	LFIN1	W,BIC			YES, FINISHED, RESTORE DATA IN C-REG
48	0055	0041	0 010 000 111		BRN	LFIN2		
49	0056		1 011 001 100	LGT2	PT11			PREPARE ADDRESS
50	0057		1 000 011 000		LDC8			
51	0060	0110	0 100 100 101		JSB	LDSC20		
52	0061	0071	0 011 100 111		BRN	LGT3		

MAIN UNIT LISTING (ROM A—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062		0 010 101 000	LEQLP	CXM		GET INT.FLG.VECTOR FOR TES
54	0063		1 001 001 100		PT9		PREPARE PAR.LEVEL INDICATOR TES
55	0064		0 001 100 010		P,CM1		IS IT ZERO?
56	0065	0203	1 000 001 111		BRN	LERRG	NO,TROUBLE
57	0066		0 010 101 000		CXM		YES,O.K. RESTORE C-REG
58	0067		1 010 001 100		PT10		
59	0070	0021	0 001 000 111		BRN	LEEQQ	LEVEL TEST FINISHED
60	0071		1 110 010 000	LGT3	ROM 7		
61	0072		1 001 010 100	LPRS60	YS9		SI SHIFT KEY ROUTINE FLAG SET ?
62	0073	0347	1 110 011 111		BRN	LPRS61	NO
63	0074		1 001 100 100		RS9		YES; RESET IT
64	0075		1 011 001 100		PT11		PREPARE POINTER TO LOAD SYMBOL CODE
65	0076		1 100 010 000		ROM 6		GO BACK
66	0077		1 100 010 000	LLAH1	ROM 6		GO TO LABEL HEAD ROUTINE BEGINNIN
67	0100		1 010 101 000	LDSC1	MTC		GET INT. FLG. VECTOR FOR UPDATE
68	0101		1 001 001 100		PT9		PREPARE LEVEL INDICATOR FOR INCREMENT
69	0102		0 111 100 010		P,CP1C		INCREMENT LEVEL INDICATOR BY ONE
70	0103		0 110 001 110		W,CTA		COPY IT INTO A-REG (FOR ADDR. GEN.)
71	0104		0 010 101 000	LDSC2	CXM		STORE IT BACK IN M-REG
72	0105		0 100 001 110		W,SLA		PREPARE LEVEL IND. FOR ADDRESS
73	0106		0 100 001 110		W,SLA		
74	0107		1 110 101 110		W,AXC		
75	0110		1 010 001 100		PT10		
76	0111		0 011 010 010	LDSC20	WP,ZTC		
77	0112		0 111 101 110		W,CP1C		
78	0113		0 011 011 110		S,ZTC		
79	0114		1 100 001 100		PT12		
80	0115		1 111 011 000		LDC15		
81	0116		1 001 110 000		ATDS		
82	0117		1 011 100 100		RS11		
83	0120		1 100 001 100		PT12		
84	0121		0 000 110 000		RETURN		END OF THE SUBROUTINE
85	0122		1 000 010 100	LCODZ	YS8		CHECK IF RIGHT PAR.FLG IS SET
86	0123	0376	1 111 111 011		BRN	LCOD0	
87	0124		0 100 000 100		SS4		YES,SET RIGHT PAR.AUX.FLAG
88	0125	0376	1 111 111 011		BRN	LCOD0	NO,NORMAL V KEY PATH
89	0126		0 011 000 100	LPAR6	SS3		SET OP.FLAG
90	0127	0322	1 101 001 011		BRN	LPAR3	
91	0130		0 101 110 010	LPER5	WP,CM1C		IS IT TWO?
92	0131	0054	0 010 110 011		BRN	LFIN1	NO; FINISHED
93	0132		1 011 000 100	LPER6	SS11		YES; SET SINGLE OP. FLAG
94	0133		1 000 010 000		ROM 4		GO TO MULTIPLICATION ROUTINE LMUL3
95	0134		1 010 101 000	LRIPD	MTC		GET INT. FLG. VECTOR FOR UPDATE
96	0135		0 110 001 110		W,CTA		COPY IN INTO A-REG
97	0136		1 001 001 100		PT9		PREPARE LEVEL INDICATOR
98	0137	0240	1 010 000 011		BRN	LRIPX	TECHNOCAL BRANCH
99	0140		1 100 010 000	LGRT2	ROM 6		
100	0141		1 100 101 000	LRIPY	DNR		
101	0142		1 011 111 000		DSTC		READ DATA OUT OF THE STACK
102	0143		0 100 101 000		CTS		SEND IT TO D-REG
103	0144	0153	0 110 110 001		JSB	LDSC3	PREPARE OP.CODE STACK
104	0145		1 001 001 110		W,SRC		OP.CODE STACK
105	0146		1 001 001 110		W,SRC		UP
106	0147		1 011 110 000		DTDS		STORE UPDATED OP. CODE STACK IN D.S.
107	0150		1 010 101 000		MTC		GET INT. FLG. VECTOR FOR UPDATE
108	0151		1 110 110 010		WP,AXC		GET OLD OP. CODE INTO INT. FLG. VECTOR
109	0152		0 010 101 000		CXM		STORE UPDATED INT. FLG. VECTOR
110	0153	0054	0 010 110 011		BRN	LFIN1	GO TO ROUTINE CLOSING PART
111	0154		0 011 001 110	LDSC3	W,ZTC		
112	0155		0 111 101 110		W,CP1C		

MAIN UNIT LISTING (ROM A—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	0156		1 111 011 000	LDC15		
114	0157		1 001 110 000	ATDS		CALL D.S.
115	0160		1 011 100 100	RS11		RESET FLAG
116	0161		1 011 111 000	DSTC		READ CURRENT OP. CODE STACK TO C-REG
117	0162		0 110 001 110	W,CTA		OP. CODE STACK TO A-REG FOR UPDATE
118	0163	0246	1 010 011 011	BRN	LDSC30	
119	0164		0 000 010 000	ROM 0		
120	0165		1 100 010 000	ROM 6		
121	0166		0 100 001 100	PT4		SET POINTER FOR "NOTE 09"
122	0167	0377	1 111 111 111	BRN	LERRZ	
123	0170		1 100 101 000	DNR		
124	0171		0 100 101 000	CTS		COPY Y BACK TO D-REG
125	0172		0 001 111 010	XS,CM1		CHECK EXPONENT SIGN
126	0173	0254	1 010 110 011	BRN	LERR32	NOT INTEGER, ERROR
127	0174		0 001 001 100	PT1		CARRY, POSITIVE EXP.COUTINUE
128	0175		0 001 100 010	P,CM1		CHECK EXPONENTS MOST SIGNIFICANT DIGIT
129	0176	0010	0 000 100 011	BRN	LGO01	NONZERO, EVEN INTEGER
130	0177		0 110 001 010	X,CTA		ZERO, TEST CONTINUE, COPY EXP TO A-REG
131	0200		0 011 001 010	X,ZTC		
132	0201		1 100 001 100	PT12		PREPARE PROPER POINTER
133	0202	0227	1 001 011 111	BRN	LERR55	
134	0203		0 010 101 000	CXM		RESTORE INT.FLG.VECTOT IN M-REG
135	0204	0245	1 010 010 111	BRN	LEEEE	
136	0205		0 010 001 110	W,BTC		RESTORE DATA IN C-REG
137	0206		0 100 000 100	SS4		SET AUX. FLAG
138	0207		0 000 010 000	ROM 0		GO TO MULTIPLICATION ROUTINE (LMUL2)
139	0210	0077	0 011 111 111	BRN	LLAH1	GO TO LABEL HEAD ROUTINE BEGINNIN
140	0211		0 000 000 000	DUMMY		
141	0212		0 000 111 100	PLS		
142	0213		0 100 011 000	LDC4		%
143	0214		1 111 011 000	LDC15		BLANK
144	0215		0 000 010 000	ROM 0		
145	0216		0 011 001 100	PT3		SET POINTER FOR "NOTE 08"
146	0217	0377	1 111 111 111	BRN	LERRZ	
147	0220	0043	0 010 001 111	BRN	LPER3	
148	0221		1 011 010 100	YS11		IS IT SINGLE OP. OPERATION?
149	0222	0170	0 111 100 011	BRN	LERR31	NO, IT IS X Y
150	0223	0166	0 111 011 011	BRN	LERR36	YES,IT IS LG OR LOG ROUTINE
151	0224	0371	1 111 100 111	BRN	LSER1	GOTO NOTE 05 MESSAGE
152	0225		0 101 000 100	SS5		SET AUXILIARY FLAG
153	0226		1 110 010 000	ROM 7		SWITCH TO ROM 7
154	0227		0 000 011 100	PRS		POINTER DECREMENT
155	0230		1 101 101 010	X,AM1A		DECREMENT EXPONENT
156	0231	0227	1 001 011 111	BRN	LERR55	NO CARRY, CONTINUE IN DECR
157	0232		0 001 110 010	WP,CM1		CARRY, POINTER ADJUSTED, CHECK TAIL
158	0233	0254	1 010 110 011	BRN	LERR32	NONZERO TAIL; NOT INTEGER
159	0234		0 000 111 100	PLS		CARRY, ZERO TAIL; INTEGER, CHECK CONT
160	0235	0271	1 011 100 111	BRN	LERR34	
161	0236	0122	0 101 001 011	LCODW BRN	LCODZ	
162	0237	0140	0 110 000 011	LGRT1 BRN	LGRT2	
163	0240		0 101 100 010	LRIPX P,CM1C		DECREMENR LEVEL INDICATOR
164	0241	0103	0 100 010 001	JSB	LDSC2	STORE IT AND CALL D.S.STACK
165	0242	0141	0 110 000 111	BRN	LRIPY	
166	0243		0 000 000 000	DUMMY		
167	0244	0322	1 101 001 011	LPRXM BRN	LPAR3	GO TO LPAR3
168	0245		1 110 010 000	LEEEE ROM 7		
169	0246		0 100 001 110	LDSC30 W,SLA		
170	0247		0 100 001 110	W,SLA		DOWN
171	0250		0 011 001 100	PT3		PREPARE OP. CODE MANIPULATION
172	0251		0 000 110 000	RETURN		END OF THE SUBROUTINE

MAIN UNIT LISTING (ROM A—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 011 001 110	LCLOM	W,ZTC		SET ZERO FOR RESULT
174	0253		0 100 010 000	LCLOD	ROM 2		
175	0254		1 000 101 110	LERR32	W,BXC		SEND EXP.TO B ; MANT. TO C-REG
176	0255		0 100 101 000		CTS		STORE MANTISSA IN STACK
177	0256	0216	1 000 111 011		BRN	LERB	
178	0257		0 111 110 000	LCLA6	CTT		COPY IT INTO T-REG
179	0260		1 000 000 000		TTP		LOAD ZEROS FROM T-REG TAIL INTO PRGM.COUNTER
180	0261	0165	0 111 010 111		BRN	LCLA8	
181	0262	0126	0 101 011 011	LIMPA	BRN	LPAR6	GO TO LEFT PAR.ROUTINE
182	0263		1 011 010 100		YS11		IS IT SINGLE OP. OPERATION?
183	0264	0364	1 111 010 011		BRN	LERR25	NO; GO TO CHECK VALUE OF Y
184	0265		1 000 010 100		YS8		IS IT LOG?
185	0266	0275	1 011 110 111		BRN	LERR24	NO, SET ONE FOR RESULT
186	0267		1 101 001 100		PT13		SET POINTER FOR "NOTE 12"
187	0270	0363	1 111 001 111		BRN	LERR39	
188	0271		0 101 100 010	LERR34	P,CM1C		DECREMENT LAST INT. DIGIT
189	0272	0003	0 000 001 111		BRN	LERR33	NO CARRY TEST CONTINUE
190	0273	0010	0 000 100 011		BRN	LGO01	CARRY EVEN, GO TO NORMAL OP.
191	0274	0014	0 000 110 011	LCLAC	BRN	LCLA3	
192	0275		0 011 001 110	LERR24	W,ZTC		
193	0276		1 100 001 100		PT12		
194	0277		0 001 011 000		LDC1		
195	0300	0253	1 010 101 111		BRN	LCLOD	
196	0301		0 000 110 100	LPARZ	CLS		CLEAR ALL STATUS BITS
197	0302		1 100 101 000		DNR		PREPARE D-REG RESET
198	0303		0 011 001 110		W,ZTC		CLEAR C-REG
199	0304		0 100 101 000		CTS		ZERO TO D-REGISTER
200	0305		0 010 000 100		SS2		SET LEFT PAR. FLG.
201	0306		0 011 000 100		SS3		SET OP. FLAG
202	0307	0042	0 010 001 011		BRN	LFIN3	GO TO ROUTINE EXIT
203	0310		0 011 010 100	LPAR2	YS3		CHECK OP. FLAG
204	0311	0205	1 000 010 111		BRN	LIMP1	NOT SET; GO TO IMPLY MULTIPLICATION
205	0312		0 010 010 100		YS2		SET, CHECK LEFT PAR. FLAG
206	0313	0126	0 101 011 011		BRN	LPAR6	NOT SET,GO TO PRINT DATA
207	0314		1 010 101 000		MTC		SET, NESTED PARENTHESES, GET IN. FLG. VECTOR
208	0315		0 011 001 100		PT3		GENERATE OP. CODE TO BE STORED
209	0316		0 000 011 000		LDC0		
210	0317		0 001 011 000		LDC1		SET ADD. AS OP CODE
211	0320		0 010 101 000		CXM		STORE NEW INT. FLG. VECTOR IN M-REG
212	0321		0 000 101 110		W,ZTB		SET "ZERO" FOR DATA TO BE STORED
213	0322	0077	0 100 000 001	LPAR3	JSB	LDSC1	GO TO INCR. LEV. IND. AND SEND DATA TO STACK
214	0323		1 100 101 000		DNR		RESTORE DATA IN C REG
215	0324		1 011 110 000		DTDS		STORE DATA IN D.S. STACK
216	0325	0153	0 110 110 001		JSB	LDSC3	PREPARE OP. CODE STACK
217	0326		1 010 101 000		MTC		INT. FLG. VECTOR OUT
218	0327		1 001 010 010		WP,SRC		PREPARE OP. CODE TO BE EXECUTED
219	0330		1 001 010 010		WP,SRC		TO BE STACKED
220	0331		1 110 101 010		X,AXC		ADD NEW OP. CODE TO STACK
221	0332		1 110 111 010		XS,AXC		NEW OP. CODE STACK IN A-REG
222	0333		0 011 010 010		WP,ZTC		CLEAR OP. CODES IN INT. FLG. VECTOR
223	0334		0 010 101 000		CXM		STORE NEW INT. FLG. VECTOR IN M-REG
224	0335		1 110 101 110		W,AXC		NEW OP. CODE STACK IN C-REG
225	0336		1 011 110 000		DTDS		STORE NEW OP. CODE STACK IN D.S.
226	0337		1 101 001 100		PT13		
227	0340		1 111 011 000	LPAR4	LDC15		BLANKS
228	0341		1 111 101 100		YP15		
229	0342	0340	1 110 000 011		BRN	LPAR4	
230	0343		0 111 110 000		CTT		GET BLANKS INTO T-REG
231	0344		1 011 001 100		PT11		
232	0345		0 010 011 000		LDC2		LOAD CODE FOR SYMBOL (

MAIN UNIT LISTING (ROM A—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
223	0346	0225	1 001 010 111		BRN	LPAR5	GO TO FORCED PRINT ROUTINE
234	0347		0 001 100 100	LPRS61	RS1		
235	0350		1 010 100 100		RS10		
236	0351	0164	0 111 010 011		BRN	LPRS62	
237	0352		0 101 110 010	LPER4	WP,CM1C		IS IT ONE?
238	0353	0130	0 101 100 011		BRN	LPER5	NO, CONTINUE IN TEST
239	0354	0132	0 101 101 011		BRN	LPER6	YES, GO TO COMMON PATH
240	0355		0 011 001 110	LERR26	W,ZTC		
241	0356		0 101 101 110		W,CM1C		
242	0357		0 011 011 110		S,ZTC		
243	0360		1 000 100 100		RS8		
244	0361		0 000 001 100		PT0		
245	0362		1 110 010 000		ROM 7		
246	0363		1 110 010 000	LERR39	ROM 7		
247	0364		1 100 101 000	LERR25	DNR		GET Y FOR TEST
248	0365		0 001 111 110		S,CM1		
249	0366	0355	1 110 110 111		BRN	LERR26	Y IS NEGATIVE
250	0367		1 000 100 100		RS8		TO AVOID COLISION WITH RPAR.ROUTINE
251	0370	0252	1 010 101 011		BRN	LCLOM	
252	0371		1 110 010 000	LSER1	ROM 7		SWITCH TO ROM 7
253	0372		1 000 001 100	LCLA20	PT8		PREPARE TRUNCATION
254	0373		0 011 010 010		WP,ZTC		TRUNCATE FRACTIONAL PART OF LABEL NUMBER
255	0374		0 110 001 110	LCLA5	W,CTA		STORE MASTER LABEL IN A-REG
256	0375	0257	1 010 111 111		BRN	LCLA6	
257	0376		1 000 010 000	LCOD0	ROM 4		
258	0377		1 110 010 000	LERRZ	ROM 7		

MAIN UNIT LISTING (ROM A—4)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
2	0000		0 000 000 000		DUMMY		
3	0001		0 000 000 000		DUMMY		
4	0002		0 010 101 000	LEQL19	CXM		PREPARE OP.CODE SET
5	0003		0 111 101 010		X,CP1C		SET OP.CODE FOR + INSTEAD
6	0004		0 010 101 000		CXM		RESTORE C AND M-REGISTERS
7	0005		1 011 001 100		PT11		PREPARE SYMBOL CODE LOAD
8	0006	0126	0 101 011 011		BRN	LEQL10	GO TO SUBTOTAL ROUTINE
9	0007		1 010 010 000	LXTY2	ROM 5		GO TO START POINT OF THE EXPO-BLOCK
10	0010		0 001 011 000	LFLG5	LDC1		SET CONDITION NOT MET FLAG
11	0011		0 010 101 000	LFLG6	CXM		STORE UPDATED INT. FLG. VECTOR
12	0012	0036	0 001 111 011		BRN	LFLG3	
13	0013		1 000 101 110	LINTE3	W,BXC		SAVE DATA TEMPORARILY IN B-REG
14	0014	0046	0 010 011 011		BRN	LFLG9	GO TO COMMON PART WITH OTHER PRGM. FLG. ROUTINES
15	0015		0 101 000 000	LCLA10	IS2		
16	0016		0 010 001 000		XOR		COMPARE WITH MASTER MASK
17	0017		1 001 000 000		IS1		
18	0020		0 111 111 000		TTC		RESULT OF COMPARISON INTO C-REG
19	0021	0216	1 000 111 011		BRN	LCLA11	
20	0022	0054	0 010 110 011	LEQLC	BRN	LEQL8	GO TO CHECK SYNTAX
21	0023		0 000 011 000	LFLG1	LDC0		RESET DIGIT INTERPRETATION CODE TO ZERO
22	0024		0 010 101 000		CXM		STORE UPDATED INT. FLG. VECTOR IN M-REG
23	0025	0077	0 100 000 001		JSB	LADDR1	GET PROGRAM FLAG VECTOR
24	0026		0 000 011 100	LFLG2	PRS		POINTER DECREMENT
25	0027		1 101 101 010		X,AM1A		DECREMENT ENTERED DIGIT TO CHECK ITS VALUE
26	0030	0026	0 001 011 011		BRN	LFLG2	NO CARRY; CONTINUE IN POINTER ADJUSTMENT
27	0031		1 000 101 110	LFLG7	W,BXC		RESTORE DATA IN B-REG; PRGM. FLG. VECTOR TO C-REG
28	0032		1 001 010 100		YS9		CHECK AUXILIARY FLAG
29	0033	0041	0 010 000 111		BRN	LFLG8	GO TO PROGRAM FLAG INTERROGATION
30	0034		0 001 011 000		LDC1		SET PROGRAM FLAG
31	0035		1 011 110 000		DTDS		STORE UPDATED PRGM. FLG. VECTOR
32	0036		1 001 100 100	LFLG3	RS9		RESET AUXILIARY FLAG IF SET
33	0037		0 010 001 110		W,BTC		RESTORE DATA IN C-REG
34	0040		0 010 010 000		ROM 1		GO TO SUPERVISOR LSUPB
35	0041		0 110 100 010	LFLG8	P,ZMC		INTERROGATE FLAG VALUE
36	0042	0045	0 010 010 111		BRN	LFLG4	ZERO ; CONDITION NOT MET
37	0043		0 000 011 000		LDC0		RESET INTERROGATED PROGRAM FLAG
38	0044		1 001 000 100		SS9		ONE ; CONDITION MET ; INDICATE THIS SITUATION
39	0045		1 011 110 000	LFLG4	DTDS		STORE UPDATED PRGM. FLG. VECTOR
40	0046		1 010 101 000	LFLG9	MTC		PREPARE INT. FLG. VECTOR
41	0047		0 110 001 100		PT6		PREPARE PROGRAM JUMP CONDITION FLAG
42	0050		1 001 010 100		YS9		CHECK CONDITION
43	0051	0010	0 000 100 011		BRN	LFLG5	NOT MET
44	0052		0 000 011 000		LDC0		MET ; SET CONDITION MET FLAG
45	0053	0011	0 000 100 111		BRN	LFLG6	STORE UPDATED INT. FLG. VECTOR
46	0054		0 011 010 100	LEQL8	YS3		IS OP. FLAG SET?
47	0055	0361	1 111 000 111		BRN	LEQL9	NO, NORMAL PATH
48	0056	0002	0 000 001 011		BRN	LEQL19	YES, "SUBTOTAL" PATH
49	0057		0 100 010 000	LCLOK	ROM 2		
50	0060		0 101 110 010	LTWO6	WP,CM1C		IS OP. CODE EQUAL TO 6?
51	0061	0353	1 110 101 111		BRN	LTWO7	NO; CONTINUE IN THE TEST
52	0062	0065	0 011 011 001		JSB	LCAL1	CALL D.S. AND GET DATA

MAIN UNIT LISTING (ROM A—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0063		1 100 101 000		DNR			RESTORE OLD DATA IN C-REG
54	0064		0 100 101 000		CTS			COPY IT BACK TO STACK
55	0065	0264	1 011 010 011		BRN	LLDD2		GO TO STORE DATA
56	0066		0 010 001 110	LCAL1	W,BTC			ADDRESS INTO C-REG
57	0067		1 001 110 000		ATDS			CALL D.S.
58	0070		1 011 010 100		YS11			CHECK IF FLAG
59	0071	0371	1 111 100 111		BRN	LERR		NO; TROUBLE; D.S. NOT FOUND
60	0072		1 011 111 000	LCAL3	DSTC			READ D.S. INTO C-REG
61	0073		1 000 101 110		W,BXC			SAVE DATA IN B-REG
62	0074		1 011 100 100		RS11			YES, O.K.
63	0075		0 110 010 100		YS6			IS THIS AN ACC ROUTINE ?
64	0076	0125	0 101 010 111		BRN	LRET1		NO, GO TO SUBROUTINE END
65	0077	0365	1 111 010 111		BRN	LCAL0		YES, GO TO DATA EXCHANGE
66	0100		0 011 001 110	LADDR1	W,ZTC			
67	0101		0 111 101 110		W,CP1C			
68	0102		1 100 001 100		PT12			
69	0103		1 111 011 000		LDC15			
70	0104		0 111 011 000		LDC7			
71	0105		1 001 110 000		ATDS			
72	0106	0072	0 011 101 011		BRN	LCAL3		
73	0107		1 000 101 110	LYLE3	W,BXC			SAVE DATA TEMPORARILY IN B-REG
74	0110	0077	0 100 000 001		JSB	LADDR1		GET PROGRAM FLAG VECTOR
75	0111		1 101 001 100		PT13			SET POINTER TO L.E. FLAG LOCATION
76	0112	0031	0 001 100 111		BRN	LFLG7		
77	0113		0 101 110 010	LTWO17	WP,CM1C			IS OP. CODE EQUAL TO 17?
78	0114	0373	1 111 101 111		BRN	LTWO18		NO; CONTINUE IN TEST
79	0115	0065	0 011 011 001		JSB	LCAL1		CALL D.S. AND GET STORED DATA
80	0116	0224	1 001 010 011		BRN	LPOW3		YES; GO TO XY ROUTINE START
81	0117		0 111 101 010	LCODX	X,CP1C			SET CURRENT OP.CODE TO ONE
82	0120	0162	0 111 001 011		BRN	LCODA		GO TO EXECUTE PREVIOUSLY KEYED OPERATION
83	0121		0 110 101 000	LSAV9	STA			OLD RESULT INTO A-REG
84	0122		1 110 101 110		W,AXC			TEMPORARILY EXCHANGED
85	0123		0 100 101 000		CTS			COPIED BACK INTO STACK
86	0124		1 110 101 110	LSAV8	W,AXC			DATA OUT OF D.S. OR STACK INTO A-REG.
87	0125		0 000 110 000	LRET1	RETURN			END OF THE SUBROUINE
88	0126		1 011 011 000	LEQL10	LDC11			R
89	0127		0 000 010 000	LEQL7	ROM 0			
90	0130		0 101 110 010	LTWO9	WP,CM1C			IS OP. CODE EQUAL TO 9
91	0131	0153	0 110 101 111		BRN	LTWO10		NO CONTINUE TEST
92	0132		0 110 000 100		SS6			YES, IT IS L X ROUTINE; SET L-FLAG
93	0133	0065	0 011 011 001	LRCM2	JSB	LCAL1		CALL D.S. TO GET STORED DATA
94	0134		0 010 001 110	LMUL3	W,BTC			RESTORE DATA IN C-REGISTER
95	0135		0 011 001 100		PT3			SET MULTIPLICATION FLAG
96	0136		0 010 101 010		X,ZMCC			COMPLEMENT OF THE EXPONENT IN C-REG
97	0137	0260	1 011 000 101	LLDV2	JSB	LSAV2		
98	0140		0 000 000 000	LDIV3	NOP			WAIT FOR OTHERS
99	0141		0 101 001 010		X,AMCC			CALCULATE RESULT EXPONENT
100	0142		0 000 101 110		W,ZTB			PREPARE B-REG
101	0143	0247	1 010 011 111		BRN	LDIV11		
102	0144		0 101 110 010	LTWO3	WP,CM1C			IS OP. CODE EQUAL TO 3?
103	0145	0147	0 110 011 111		BRN	LTWO4		NO; CONTINUE TEST
104	0146	0134	0 101 110 011		BRN	LMUL3		YES; GO TO MULTIPLICATION ROUTINE (x)
105	0147		0 101 110 010	LTWO4	WP,CM1C			IS OP. CODE EQUAL TO 4?
106	0150	0222	1 001 001 011		BRN	LTWO5		NO; CONTINUE TEST
107	0151		1 100 001 100	LRCV2	PT12			YES; SET POINTER TO PROPER VALUE
108	0152	0177	0 111 111 111		BRN	LDIV9		
109	0153		0 101 110 010	LTWO10	WP,CM1C			IS OP. CODE EQUAL TO 10?
110	0154	0217	1 000 111 111		BRN	LTWO11		NO; CONTINUE TEST
111	0155		0 110 000 100		SS6			YES; IT IS L-ROUTINE,SET L-FLAG
112	0156	0065	0 011 011 001		JSB	LCAL1		CALL D.S. TO GET STORED DATA

MAIN UNIT LISTING (ROM A—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0157		1 100 001 100		PT12			SET PROPER POINTER VALUE FOR DIVISION
114	0160	0177	0 111 111 111		BRN	LDIV9		GO TO RESTORE DIVISOR IN C-REG
115	0161	0015	0 000 110 111	LCLAE	BRN	LCLA10		
116	0162		0 010 101 000	LCODA	CXM			STORE UPDATED INT. FLG. VECTOR IN M-REG
117	0163		1 010 101 000		MTC			PREPARE NEW INT. FLG. VECTOR FOR TEST
118	0164		1 001 001 110		W,SRC			OLD OP. CODE ALIGNMENT SHIFT
119	0165		1 001 001 110		W,SRC			OLD OP. CODE ALIGNMENT SHIFT
120	0166		0 001 001 100		PT1			PREPARE OLD OP. CODE (TO BE EXECUTED) FOR TEST
121	0167		0 110 100 100		RS6			
122	0170		0 101 110 010	LTWO0	WP,CM1C			IS OP. CODE EQUAL TO 0?
123	0171	0250	1 010 100 011		BRN	LTWO1		NO, CONTINUE TEST
124	0172		0 010 001 110	LCLOH	W,BTC			RESTORE DATA IN C-REG
125	0173	0057	0 010 111 111		BRN	LCLOK		GO TO COPY IT INTO STACK
126	0174		0 101 110 010	LTWO15	WP,CM1C			IS OP. CODE EQUAL TO 15?
127	0175	0212	1 000 101 011		BRN	LTWO16		NO; CONTINUE TEST
128	0176	0133	0 101 101 111		BRN	LRCM2		GO TO COMMON PART OF ROUTINES
129	0177		0 010 001 110	LDIV9	W,BTC			
130	0200	0137	0 101 111 111		BRN	LLDV2		
131	0201	0013	0 000 101 111	LINTEA	BRN	LINTE3		
132	0202	0107	0 100 011 111	LYLEA	BRN	LYLE3		
133	0203		0 101 111 010	LSUM5	XS,CM1C			RESTORE RESULT EXPONENT IN C-REG
134	0204		0 101 111 010		XS,CM1C			RESTORE RESULT EXPONENT IN C-REG
135	0205		1 011 101 010		X,ZTA			CANCEL OFFSET INDICATOR
136	0206		1 101 011 110		S,AMCA			CHECK SIGNS OF BOTH MANITSSAS
137	0207		1 001 111 110		S,AM1			EQUAL?
138	0210	0306	1 100 011 011		BRN	LSUM6		NO; GO TO SUBTRACTION
139	0211	0305	1 100 010 111		BRN	LSUM8		YES; GO TO ADDITION
140	0212		0 101 110 010	LTWO16	WP,CM1C			IS OP. CODE EQUAL TO 16?
141	0213	0113	0 100 101 111		BRN	LTWO17		NO; CONTINUE TEST
142	0214	0065	0 011 011 001		JSB	LCAL1		CALL D.S. TO GET STORED DATA
143	0215	0151	0 110 100 111		BRN	LRCV2		GO TO COMMON PART OF ROUTINES
144	0216		1 100 010 000	LCLA11	ROM 6			
145	0217		0 101 110 010	LTWO11	WP,CM1C			IS OP. CODE EQUAL TO 11?
146	0220	0362	1 111 001 011		BRN	LTWO12		NO; CONTINUE IN TEST
147	0221	0362	1 111 001 011		BRN	LTWO12		NO; CONTINUE IN TEST
148	0222		0 101 110 010	LTWO5	WP,CM1C			IS OP. CODE EQUAL TO 5?
149	0223	0060	0 011 000 011		BRN	LTWO6		NO; CONTINUE IN TEST
150	0224		1 100 101 000	LPOW3	DNR			XY - ROUTINE START; PREPARE MANTISSA, EXPONENT
151	0225		1 000 101 110		W,BXC			
152	0226		0 100 101 000		CTS			MANTISSA IN A,C,D-REG
153	0227		0 010 001 110		W,BTC			EXPONENT IS IN B,C-REG
154	0230		1 000 000 100		SS8			
155	0231		0 010 000 100	LXTY1	SS2			XY-ROUTINE EXECUTION START POINT
156	0232		1 100 001 100		PT12			SET POINTER TO PROPER VALUE
157	0233	0007	0 000 011 111		BRN	LXTY2		
158	0234		0 100 001 110	LPQO11	W,SLA			
159	0235		1 010 010 110	LPQO12	MS,SRB			
160	0236		1 000 101 110		W,BXC			
161	0237	0241	1 010 000 111		BRN	LPQO16		
162	0240		0 111 111 110	LPQO15	S,CP1C			
163	0241		1 100 001 110	LPQO16	W,AMBA			
164	0242	0240	1 010 000 011		BRN	LPQO15		
165	0243		1 110 001 110		W,APBA			
166	0244		1 010 010 000	LPQO13	ROM 5			GO TO LPQO23
167	0245		1 100 001 100	LSAV1	PT12			SET PROPER POINTER VALUE
168	0246	0261	1 011 000 111		BRN	LSAV2		
169	0247		1 010 010 000	LDIV11	ROM 5			GO TO LDIV21
170	0250		0 101 110 010	LTWO1	WP,CM1C			IS OP. CODE EQUAL TO 1?
171	0251	0253	1 010 101 111		BRN	LTWO2		NO; CONTINUE TEST
172	0252	0357	1 110 111 111		BRN	LADD3		YES; GO TO ADDITION ROUTINE

MAIN UNIT LISTING (ROM A—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0253		0 101 110 010	LTWO2	WP,CM1C	IS OP. CODE EQUAL TO 2?
174	0254	0144	0 110 010 011		BRN LTWO3	NO; CONTINUE TEST
175	0255	0320	1 101 000 011		BRN LSUB3	YES; GO TO SUBTRACTION ROUTINE
176	0256		0 101 110 010	LTWO13	WP,CM1C	IS OP. CODE EQUAL TO 13?
177	0257	0347	1 110 011 111		BRN LTWO14	NO; CONTINUE TEST
178	0260	0356	1 110 111 011		BRN LRCA2	GO TO CALL DATA FROM D.S.
179	0261		0 110 010 100	LSAV2	YS6	IS THIS AN ACC ROUINE ?
180	0262	0121	0 101 000 111		BRN LSAV9	NO,GO TO GET OLD RESULT
181	0263		0 000 110 000		RETURN	YES,DO NOTHING,DATA IS READY
182	0264		0 100 010 000	LLDD2	ROM 2	GO TO COMMON STORE ROUTINE
183	0265	0023	0 001 001 111	LFLAG	BRN LFLG1	
184	0266		0 100 010 000	LCLO4	ROM 2	
185	0267		0 100 010 000	LCLO5	ROM 2	
186	0270		0 111 100 010	LDIV14	P,CP1C	
187	0271		1 100 010 110	LDIV15	MS,AMBA	
188	0272	0270	1 011 100 011		BRN LDIV14	
189	0273		1 110 010 110		MS,APBA	
190	0274		0 100 010 110		MS,SLA	
191	0275		0 000 011 100	LDIV16	PRS	
192	0276		0 000 101 100		YP0	
193	0277	0271	1 011 100 111		BRN LDIV15	
194	0300		0 110 001 110	LTNM12	W,CTA	
195	0301		0 010 001 010		X,BTC	
196	0302	0313	1 100 101 111		BRN LADD15	
197	0303		0 100 101 000	LERRR	CTS	STORE IT ALSO IN STACK
198	0304		0 000 010 000		ROM 0	CALL ERROR ROUTINE
199	0305		1 010 010 000	LSUM8	ROM 5	GO TO LMPY 26 AND THEN TO NORMALIZATION
200	0306		1 000 000 110	LSUM6	M,AMB	CHECK WHICH MANTISSA IS BIGGER
201	0307	0312	1 100 101 011		BRN LSUM7	A-REG HAS BIGGER MANTISSA; GO TO SUBTRACTION
202	0310		0 011 111 110		S,ZNCC	B-REG HAS BIGGER MANTISSA; CHANGE RESULT SIGN.
203	0311		1 100 101 110		W,AXB	INTERCHANGE A AND B-REGISTERS
204	0312		1 100 001 110	LSUM7	W,AMBA	PERFORM SUBTRACTION
205	0313		1 010 010 000	LADD15	ROM 5	GO TO NORMALIZATION ROUTINE
206	0314		0 101 110 010	LTWO8	WP,CM1C	IS OP. CODE EQUAL TO 8?
207	0315	0130	0 101 100 011		BRN LTWO9	NO; CONTINUE TEST
208	0316		0 110 000 100		SS6	YES; IT IS L-ROUTINE; SET L-FLAG
209	0317	0065	0 011 011 001	LRCS2	JSB LCAL1	CALL D.S. AND GET STORED DATA
210	0320		0 010 001 110	LSUB3	W,BTC	RESTORE DATA IN C-REGISTER
211	0321		0 011 111 110		S,ZNCC	COMPLEMENT OF THE SIGN OF DATA IN C-REG
212	0322	0244	1 010 010 101	LSUM1	JSB LSAV1	GET PREVIOUS RESULT
213	0323		0 000 101 110		W,ZTB	PREPARE B-REGISTER
214	0324		1 111 111 010		XS,AP1A	PREPARE SIGN OF EXP. IN A-REG FOR TEST
215	0325		1 111 111 010		XS,AP1A	PREPARE SIGN OF EXP. IN A-REG FOR TEST
216	0326		0 111 111 010		XS,CP1C	PREPARE SIGN OF EXP. IN C-REG FOR TEST
217	0327		0 111 111 010		XS,CP1C	PREPARE SIGN OF EXP. IN C-REG FOR TEST
218	0330		0 001 001 010		X,AMC	CHECK WHICH EXPONENT IS BIGGER
219	0331	0333	1 101 101 111		BRN LSUM2	BIGGER IS IN A-REGISTER
220	0332		1 110 101 110		W,AXC	GET DATA WITH BIGGER EXPONENT INTO A-REG
221	0333		1 110 100 110	LSUM2	M,AXC	PREPARE MANTISSA VALUE CHECK
222	0334		0 110 100 110		M,ZMC	CHECK IF MANTISSA IS ZERO
223	0335	0337	1 101 111 111		BRN LSUM3	YES
224	0336		1 110 101 110		W,AXC	NO; PUT IT BACK TO A-REG
225	0337		1 000 100 110	LSUM3	M,BXC	PREPARE MANTISSA ALIGNMENT IN B-REG
226	0340		0 001 001 010	LSUM4	X,AMC	CHECK IF OFFSET IS ZERO
227	0341	0203	1 000 001 111		BRN LSUM5	YES, MANTISSAS ARE ALIGNED FOR SUMMATION
228	0342		1 010 001 110		W,SRB	NO; ALIGNMENT SHIFT
229	0343		1 111 101 010		X,AP1A	DECREMENT OFFSET
230	0344		0 000 001 110		W,ZMB	CHECK IF MANTISSA AFTER SHIFT IS ZERO
231	0345	0203	1 000 001 111		BRN LSUM5	YES, GO DIRECTLY TO RESULT CALCULATION
232	0346	0340	1 110 000 011		BRN LSUM4	NO, CONTINUE IN OFFSET ALIGNMENT

MAIN UNIT LISTING (ROM A—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
233	0347		0 101 110 010	LTWO14	WP,CM1C			IS OP. CODE EQUAL TO 14?
234	0350	0174	0 111 110 011		BRN	LTWO15		NO; CONTINUE TEST
235	0351		0 110 100 100		RS6			YES; IT IS R-ROUTINE; RESET L-FLAG
236	0352	0317	1 100 111 111		BRN	LRCS2		GO TO CALL DATA FROM D.S.
237	0353		0 101 110 010	LTWO7	WP,CM1C			IS OP. CODE EQUAL TO 7?
238	0354	0314	1 100 110 011		BRN	LTWO8		NO; CONTINUE TEST
239	0355		0 110 000 100		SS6			YES; IT IS L ROUTINE; SET L-FLAG
240	0356	0065	0 011 011 001	LRCA2	JSB	LCAL1		CALL D.S. AND GET STORED DATA
241	0357		0 010 001 110	LADD3	W,BTC			RESTORE DATA IN C-REGISTER
242	0360	0322	1 101 001 011		BRN	LSUM1		GO TO COMMON SUMMATION BLOCK
243	0361		0 010 010 000	LEQL9	ROM 1			SWITCH TO ROM 1
244	0362		0 101 110 010	LTWO12	WP,CM1C			IS OP. CODE EQUAL TO 12?
245	0363	0256	1 010 111 011		BRN	LTWO13		NO; CONTINUE IN TEST
246	0364	0356	1 110 111 011		BRN	LRCA2		GO TO CALL DATA FROM D.S.
247	0365		1 100 101 000	LCAL0	DNR			RESTORE PREVIOUS RESULT IN C-REG
248	0366		0 100 101 000		CTS			COPY IT BACK TO STACK
249	0367		1 000 101 110	LSAV7	W,BXC			OLD RESULT TO B,DATA OUT OF D.S. TO C-REG
250	0370	0124	0 101 010 011		BRN	LSAV8		
251	0371		1 100 101 000	LERR	DNR			RESTORE OLD DATA NORIGINAL DATA)
252	0372	0303	1 100 001 111		BRN	LERRR		
253	0373	0065	0 011 011 001	LTWO18	JSB	LCAL1		CALL D.S. AND GET "NEW" DATA INTO B-REG
254	0374		1 100 101 000		DNR			PREPERE OLD DATA
255	0375		1 011 110 000		DTDS			STORE "OLD" DATA IN D.S.
256	0376	0172	0 111 101 111		BRN	LCLOH		
257	0377	0117	0 100 111 111	LCODD	BRN	LCODX		GO TO SET OP.CODE

MAIN UNIT LISTING (ROM A—5)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 110 010 000	LERR21	ROM 3		
4	0001		1 100 111 110	LLN24	S,AXB		
5	0002		1 111 111 110		S,AP1A		
6	0003		1 001 010 110		MS,SRC		
7	0004		0 100 010 010		WP,SLA		
8	0005	0022	0 001 001 011		BRN	LLN26	
9	0006		0 110 101 000	LXTY22	STA		
10	0007	0245	1 010 011 001		JSB	LMPY21	
11	0010	0263	1 011 001 111	LSBL1	BRN	LSBL2	GO TO SET BUSY LIGHT
12	0011		1 000 010 100	LAUX1	YS8		
13	0012	0102	0 100 001 011		BRN	LEXP21	
14	0013		1 011 101 110	LLN22	W,ZTA		
15	0014		1 101 000 110		M,AMCA		
16	0015	0262	1 011 001 011		BRN	LERR23	MANTISSA IS ZERO
17	0016		1 011 001 110		W,SRA		
18	0017		0 101 111 110		S,CM1C		
19	0020	0000	0 000 000 011		BRN	LERR21	MANTISSA IS NEGATIVE
20	0021		0 111 111 110	LLN25	S,CP1C		
21	0022		0 100 101 110	LLN26	W,ATB		
22	0023	0225	1 001 011 001		JSB	LECA22	
23	0024		1 101 100 010		P,AM1A		
24	0025	0021	0 001 000 111		BRN	LLN25	
25	0026		1 100 110 010		WP,AXB		
26	0027		1 110 011 110		S,APBA		
27	0030	0001	0 000 000 111		BRN	LLN24	
28	0031		0 111 001 100		PT7		
29	0032	0154	0 110 110 101		JSB	LPQO23	
30	0033		1 000 001 100		PT8		
31	0034	0234	1 001 110 101		JSB	LPMU22	
32	0035		1 001 001 100		PT9		
33	0036	0233	1 001 110 001		JSB	LPMU21	
34	0037	0375	1 111 111 001		JSB	LLNCD3	
35	0040		1 010 001 100		PT10		
36	0041	0233	1 001 110 001		JSB	LPMU21	
37	0042	0174	0 111 110 101		JSB	LLNCD2	
38	0043		1 011 001 100		PT11		
39	0044	0233	1 001 110 001		JSB	LPMU21	
40	0045	0336	1 101 111 101		JSB	LLNCD1	
41	0046	0233	1 001 110 001		JSB	LPMU21	
42	0047	0270	1 011 100 101		JSB	LLNC2	
43	0050	0233	1 001 110 001		JSB	LPMU21	
44	0051	0365	1 111 011 001		JSB	LLNC10	
45	0052		1 110 101 110		W,AXC		
46	0053		0 101 001 110		W,AMCC		
47	0054		0 000 011 010		XS,ZMB		
48	0055	0057	0 010 111 111		BRN	LLN27	
49	0056		0 101 001 110		W,AMCC		
50	0057		1 100 101 110	LLN27	W,AXB		
51	0060		0 000 011 100	LLN28	PRS		
52	0061		0 100 001 110		W,SLA		

MAIN UNIT LISTING (ROM A—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
53	0062		0 001 101 100		YP1	
54	0063	0060	0 011 000 011		BRN	LLN28
55	0064		1 110 101 110		W,AXC	
56	0065		0 110 111 110		S,ZMC	
57	0066	0070	0 011 100 011		BRN	LLN29
58	0067		0 011 100 110		M,ZNCC	
59	0070		0 111 101 010	LLN29	X,CP1C	
60	0071		1 011 001 100		PT11	
61	0072	0304	1 100 010 101		JSB	LMPY27
62	0073		1 001 010 100		YS9	
63	0074	0006	0 000 011 011		BRN	LXTY22
64	0075		0 101 010 100		YS5	
65	0076	0224	1 001 010 011		BRN	LRTN21
66	0077	0365	1 111 011 001		JSB	LLNC10
67	0100	0246	1 010 011 101		JSB	LMPY22
68	0101	0224	1 001 010 011		BRN	LRTN21
69	0102	0365	1 111 011 001	LEXP21	JSB	LLNC10
70	0103	0353	1 110 110 001		JSB	LPRE21
71	0104	0270	1 011 100 101		JSB	LLNC2
72	0105		1 011 001 100		PT11	
73	0106	0232	1 001 101 101		JSB	LPQO21
74	0107	0336	1 101 111 101		JSB	LLNCD1
75	0110		1 010 001 100		PT10	
76	0111	0232	1 001 101 101		JSB	LPQO21
77	0112	0174	0 111 110 101		JSB	LLNCD2
78	0113		1 001 001 100		PT9	
79	0114	0232	1 001 101 101		JSB	LPQO21
80	0115	0375	1 111 111 001		JSB	LLNCD3
81	0116		1 000 001 100		PT8	
82	0117	0232	1 001 101 101		JSB	LPQO21
83	0120	0232	1 001 101 101		JSB	LPQO21
84	0121	0232	1 001 101 101		JSB	LPQO21
85	0122		0 110 001 100		PT6	
86	0123		1 011 110 010		WP,ZTA	
87	0124		1 101 001 100		PT13	
88	0125		1 000 101 110		W,BXC	
89	0126		1 110 101 110		W,AXC	
90	0127		0 110 011 000		LDC6	
91	0130	0216	1 000 111 011		BRN	LEXP23
92	0131		0 010 010 100	LPRE23	YS2	
93	0132	0136	0 101 111 011		BRN	LPRE24
94	0133		1 111 101 010		X,AP1A	
95	0134		1 001 111 010	LPRE29	XS,AM1	
96	0135	0302	1 100 001 011		BRN	LPRE27
97	0136		1 100 010 110	LPRE24	MS,AMBA	
98	0137	0131	0 101 100 111		BRN	LPRE23
99	0140		1 110 010 110		MS,APBA	
100	0141		0 100 001 110		W,SLA	
101	0142		0 101 101 010		X,CM1C	
102	0143	0134	0 101 110 011		BRN	LPRE29
103	0144		1 011 001 110	LPRE25	W,SRA	
104	0145		0 011 010 010		WP,ZTC	
105	0146		1 110 101 010		X,AXC	
106	0147		0 110 111 110	LPRE26	S,ZMC	
107	0150	0154	0 110 110 011		BRN	LPRE28
108	0151		1 100 101 110		W,AXB	
109	0152		1 100 001 110		W,AMBA	
110	0153		0 011 101 110		W,ZNCC	
111	0154		1 011 001 110	LPRE28	W,SRA	
112	0155		1 000 101 110	LPQO23	W,BXC	

MAIN UNIT LISTING (ROM A—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	0156		0 011 001 110		W,ZTC	
114	0157		0 101 100 110		M,CM1C	
115	0160		0 010 010 100		YS2	
116	0161	0166	0 111 011 011		BRN	LPQO28
117	0162		0 100 011 000		LDC4	
118	0163		0 111 100 110		M,CP1C	
119	0164	0171	0 111 100 111		BRN	LPQO24
120	0165		0 110 011 000	LPQO27	LDC6	
121	0166		0 001 101 100	LPQO28	YP1	
122	0167	0165	0 111 010 111		BRN	LPQO27
123	0170		1 001 001 110		W,SRC	
124	0171		1 001 001 110	LPQO24	W,SRC	
125	0172		0 010 010 100	LNRM26	YS2	
126	0173	0224	1 001 010 011		BRN	LRTN21
127	0174		0 000 110 000		RETURN	
128	0175		0 111 001 100	LLNCD2	PT7	
129	0176		0 011 011 000	LLNC6	LDC3	
130	0177		0 011 011 000		LDC3	
131	0200		0 000 011 000		LDC0	
132	0201		1 000 011 000	LLNC7	LDC8	
133	0202		0 101 011 000		LDC5	
134	0203		0 000 011 000		LDC0	
135	0204		1 001 011 000		LDC9	
136	0205	0352	1 110 101 011		BRN	LLNC9
137	0206	0225	1 001 011 001	LEXP29	JSB	LECA22
138	0207		1 111 100 010		P,AP1A	
139	0210		0 100 101 110	LEXP22	W,ATB	
140	0211		0 101 111 110		S,CM1C	
141	0212	0206	1 000 011 011		BRN	LEXP29
142	0213		1 011 010 010		WP,SRA	
143	0214		1 110 101 110		W,AXC	
144	0215		0 100 010 110		MS,SLA	
145	0216		1 110 101 110	LEXP23	W,AXC	
146	0217		1 101 111 110		S,AM1A	
147	0220	0210	1 000 100 011		BRN	LEXP22
148	0221		1 100 101 110		W,AXB	
149	0222		1 111 100 010		P,AP1A	
150	0223	0313	1 100 110 001		JSB	LNRM21
151	0224		0 100 010 000	LRTN21	ROM 2	END OF NORMALIZATION, GO TO OVFL TEST
152	0225		1 011 010 010	LECA21	WP,SRA	
153	0226		1 101 111 110	LECA22	S,AM1A	
154	0227	0225	1 001 010 111		BRN	LECA21
155	0230		1 011 111 110		S,ZTA	
156	0231		1 110 001 110		W,APBA	
157	0232		0 000 110 000		RETURN	
158	0233		1 000 010 000	LPQO21	ROM 4	GO TO LPQO11
159	0234		1 011 001 110	LPMU21	W,SRA	
160	0235		1 000 101 110	LPMU22	W,BXC	
161	0236	0240	1 010 000 011		BRN	LPMU24
162	0237		1 110 001 110	LPMU23	W,APBA	
163	0240		0 101 111 110	LPMU24	S,CM1C	
164	0241	0237	1 001 111 111		BRN	LPMU23
165	0242		1 110 101 110		W,AXC	
166	0243		0 100 010 110		MS,SLA	
167	0244		1 110 101 110		W,AXC	
168	0245	0155	0 110 110 111		BRN	LPQO23
169	0246		0 011 001 100	LMPY21	PT3	
170	0247		0 111 001 010	LMPY22	X,APCC	
171	0250		0 101 011 110	LDIV21	S,AMCC	
172	0251	0253	1 010 101 111		BRN	LDIV22

MAIN UNIT LISTING (ROM A—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0252		0 010 111 110		S,ZMCC	
174	0253		1 100 100 110	LDIV22	M,AXB	
175	0254		1 011 101 110		W,ZTA	
176	0255		1 100 101 100		YP12	
177	0256	0305	1 100 010 111		BRN	LMPY27
178	0257		0 001 100 110		M,CM1	
179	0260	0266	1 011 011 011		BRN	LDIV23
180	0261		1 110 010 000	LERR22	ROM 7	ZERO DIVISOR DETECTED <C>=0 LNOT3
181	0262		0 110 010 000	LERR23	ROM 3	
182	0263		1 010 100 000	LSBL2	SBL	SET BUSY LIGHT
183	0264		0 110 001 110	LXTY21	W,CTA	EXPO BLOCK EXECUTION STARTING POINT
184	0265	0011	0 000 100 111		BRN	LAUX1
185	0266		1 000 110 010	LDIV23	WP,BXC	
186	0267		1 110 100 110		M,AXC	
187	0270		1 000 010 000		ROM 4	GO TO LDIV15
188	0271		1 000 100 100	LLNC2	RS8	
189	0272		0 110 011 000		LDC6	
190	0273		1 001 011 000		LDC9	
191	0274		0 011 011 000		LDC3	
192	0275		0 001 011 000		LDC1	
193	0276		0 100 011 000		LDC4	
194	0277		0 111 011 000		LDC7	
195	0300		0 001 011 000		LDC1	
196	0301	0346	1 110 011 011		BRN	LLNC8
197	0302		1 111 100 110	LPRE27	M,AP1A	
198	0303	0144	0 110 010 011		BRN	LPRE25
199	0304		1 110 001 110	LMPY26	W,APBA	
200	0305		0 101 100 010	LMPY27	P,CM1C	
201	0306	0304	1 100 010 011		BRN	LMPY26
202	0307		1 011 001 110	LMPY28	W,SRA	
203	0310		0 000 111 100		PLS	
204	0311		1 101 101 100		YP13	
205	0312	0305	1 100 010 111		BRN	LMPY27
206	0313		0 111 101 010		X,CP1C	
207	0314		1 011 111 110	LNRM21	S,ZTA	
208	0315		1 100 001 100		PT12	
209	0316		0 000 101 110		W,ZTB	
210	0317		1 001 100 010	LNRM23	P,AM1	
211	0320	0326	1 101 011 011		BRN	LNRM24
212	0321		0 100 001 110		W,SLA	
213	0322		0 101 101 010		X,CM1C	
214	0323		1 001 101 110		W,AM1	
215	0324	0317	1 100 111 111		BRN	LNRM23
216	0325		0 011 001 110		W,ZTC	
217	0326		0 100 101 010	LNRM24	X,ATB	
218	0327		1 110 001 110		W,APBA	
219	0330		1 001 111 110		S,AM1	
220	0331	0307	1 100 011 111		BRN	LMPY28
221	0332		1 110 100 110		M,AXC	
222	0333		0 110 001 110	LNRM25	W,CTA	
223	0334		0 000 101 110		W,ZTB	
224	0335		1 100 001 100	LNRM27	PT12	
225	0336	0172	0 111 101 011		BRN	LNRM26
226	0337		1 001 001 100	LLNCD1	PT9	
227	0340		0 011 011 000		LDC3	
228	0341		0 001 011 000		LDC1	
229	0342		0 000 011 000		LDC0	
230	0343		0 001 011 000		LDC1	
231	0344		0 111 011 000		LDC7	
232	0345		1 001 011 000		LDC9	

MAIN UNIT LISTING (ROM A—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
233	0346		1 000 011 000	LLNC8	LDC8	
234	0347		0 000 011 000		LDC0	
235	0350		0 101 011 000		LDC5	
236	0351		0 101 011 000		LDC5	
237	0352		0 011 011 000	LLNC9	LDC3	
238	0353	0335	1 101 110 111		BRN	LNRM27
239	0354		1 110 101 110	LPRE21	W,AXC	
240	0355		0 100 101 110		W,ATB	
241	0356		0 110 000 110		M,CTA	
242	0357		1 010 111 010		XS,CPCC	
243	0360	0136	0 101 111 011		BRN	LPRE24
244	0361		0 111 111 010		XS,CP1C	
245	0362		1 011 001 110	LPRE22	W,SRA	
246	0363		0 111 101 010		X,CP1C	
247	0364	0362	1 111 001 011		BRN	LPRE22
248	0365	0147	0 110 011 111		BRN	LPRE26
249	0366		0 011 001 110	LLNC10	W,ZTC	
250	0367		1 100 001 100		PT12	
251	0370		0 010 011 000		LDC2	
252	0371		0 011 011 000		LDC3	
253	0372		0 000 011 000		LDC0	
254	0373		0 010 011 000		LDC2	
255	0374		0 101 011 000		LDC5	
256	0375	0201	1 000 000 111		BRN	LLNC7
257	0376		0 101 001 100	LLNCD3	PT5	
258	0377	0176	0 111 111 011		BRN	LLNC6

MAIN UNIT LISTING (ROM A—6)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000		0 000 010 000	LLLL6	ROM 0	GO TO RM 0 IN "RON GROUP SWITCH MODE"
4	0001		1 000 000 100	LLN1	SS8	NOT. LOG. ROUTINE EXECUTION START
5	0002		1 001 000 100	LEXP1	SS9	EXPONENTIAL ROUTINE EXECUTION START
6	0003		0 010 000 100		SS2	SET COMMON FLAG OF EXP. ROUTINES
7	0004		1 011 000 100		SS11	SET SINGLE OPERAND OP. FLAG
8	0005		0 110 100 100		RS6	RESET DEC. POINT FLAG IF SET
9	0006		1 100 001 100		PT12	SET POINTER TO PROPER VALUE
10	0007		1 010 010 000		ROM 5	GO TO START POINT OF THE EXPO BLOCK
11	0010		1 001 000 100	LPREP	SS9	SET SHIFT KEY ROUTINE AUX. FLAG
12	0011	0024	0 001 010 011		BRN LPREP1	
13	0012	0101	0 100 000 111	LLOGA	BRN LLOGB	ROUTINE START
14	0013	0133	0 101 101 111	LLNA	BRN LLNB	ROUTINE START
15	0014	0007	0 000 100 001	LOXXA	JSB LPREP	PREPARE DATA FOR PRINT
16	0015	0112	0 100 101 011		BRN LOXXB	CONTINUE
17	0016		0 100 100 000	LEXCH	RMGRC	EXCH() ROUTINE BEGINNING POINT
18	0017		1 110 010 000	LERRR	ROM 7	GO TO ERROR MESSAGE ROUTINE
19	0020	0055	0 010 110 111		BRN LGT1	GRAND TOTAL
20	0021		1 110 010 000	LGTR	ROM 7	AFTER RETURN FROM DATA ADJUSTMENT LOAD SYMBOL CODES
21	0022	0067	0 011 011 111	LTAB1	BRN LTAB2	CALL TABLE FUNCTION
22	0023	0106	0 100 011 011		BRN LDONT	DO NOTHING
23	0024		1 110 010 000	LPREP1	ROM 7	
24	0025		0 000 000 000		DUMMY	
25	0026		0 011 111 110	LCHS1	S,ZNCC	CHANGE SIGN OF MANTISSA
26	0027		0 000 000 000		NOP	TO ELIMINATE POSSIBLE CARRY
27	0030	0106	0 100 011 011		BRN LDONT	GO TO SUPERVISOR
28	0031	0354	1 110 110 101	LALF8	JSB LPRINT	PRINT ALPHA MESS,RESTORE DATA
29	0032	0106	0 100 011 011		BRN LDONT	DO NOTHING
30	0033	0106	0 100 011 011		BRN LDONT	DO NOTHING
31	0034	0106	0 100 011 011		BRN LDONT	DO NOTHING
32	0035		0 010 101 000	LIOC3	CXM	STORE UPDATED INT. FLG. VECTOR
33	0036	0106	0 100 011 011		BRN LDONT	DO NOTHING
34	0037		0 010 100 000	LFBLK	RMGRB	CALL F. BLOCK SOFTWARE
35	0040	0007	0 000 100 001	LEXPA	JSB LPREP	PREPARE DATA FOR PRINT
36	0041	0146	0 110 011 011		BRN LEXPB	
37	0042	0106	0 100 011 011		BRN LDONT	DO NOTHING
38	0043	0106	0 100 011 011		BRN LDONT	DO NOTHING
39	0044	0106	0 100 011 011		BRN LDONT	DO NOTHING
40	0045		0 011 010 000	LMANZ	TKRA	ACCEPT KEYCODE
41	0046		0 100 100 000	LRECL	RMGRC	RECALL() ROUTINE BEGINNING POINT
42	0047	0017	0 000 111 111		BRN LERRR	D.S. SOFTWARE NOT PRESENT
43	0050	0007	0 000 100 001	LTWWA	JSB LPREP	PREPARE DATA FOR PRINT
44	0051	0121	0 101 000 111		BRN LTWWB	CONTINUE
45	0052	0106	0 100 011 011		BRN LDONT	DO NOTHING
46	0053	0106	0 100 011 011		BRN LDONT	DO NOTHING
47	0054	0106	0 100 011 011		BRN LDONT	DO NOTHING
48	0055		0 110 010 000	LGT1	ROM 3	
49	0056		0 100 100 000	LSTOR	RMGRC	STORE() ROUTINE BEGINNING POINT
50	0057	0017	0 000 111 111		BRN LERRR	D.S. SOFTWARE NOT PRESENT
51	0060	0303	1 100 010 001	LALFB	JSB LGNT2	GET TWO INSTR. TO BUILD RIGHT PART
52	0061	0155	0 110 110 111		BRN LALF2	

MAIN UNIT LISTING (ROM A—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0062	0106	0 100 011 011		BRN	LDONT	DO NOTHING	
54	0063	0106	0 100 011 011		BRN	LDONT	DO NOTHING	
55	0064		0 100 100 000	LLIST	RMGRC		LIST OR CLEAR DATA BEGINNING POINT	
56	0065	0017	0 000 111 111		BRN	LERRR	D.S. SOFTWARE NOT PRESENT	
57	0066	0106	0 100 011 011		BRN	LDONT	DO NOTHING	
58	0067		0 010 100 000	LTAB2	RMGRB		GO TO F.BLK.SOFTWARE	
59	0070		0 000 000 000	LSHFT2	NOP			
60	0071		1 010 000 100		SS10		SET SHIFT KEY FLAG	
61	0072	0106	0 100 011 011		BRN	LDONT	DO NOTHING	
62	0073	0106	0 100 011 011		BRN	LDONT	DO NOTHING	
63	0074		0 111 101 110	LTWW2	W,CP1C		GEN EXP. VALOE 1	
64	0075	0117	0 100 111 111		BRN	LOXXD		
65	0076	0142	0 110 001 011	LIOC1	BRN	LIOC2	I/O CALL BEGINNING	
66	0077		0 000 110 000		RETURN		END OF SUBROUTINE	
67	0100	0152	0 110 101 011	LLAHB	BRN	LINCR	GO TO LABEL HEAD ROUT.EXECUTION	
68	0101	0007	0 000 100 001	LLOGB	JSB	LPREP	PREPARE DATA FOR PRINT	
69	0102		1 000 011 000		LDC8		LOAD SYMBOL CODE FOR LOG	
70	0103	0354	1 110 110 101		JSB	LPRINT	PRINT AND RESTORE DATA IN C-REG	
71	0104		0 101 000 100		SS5		DEC. LOG ROUTINE EXECUTION START	
72	0105	0001	0 000 000 111		BRN	LLN1	GO TO COMMON PART WITH LN ROUTINE	
73	0106		0 000 010 000	LDONT	ROM 0		GO TO SUPERVISOR	
74	0107		0 101 100 010	LRET1	P,CM1C		DECREMENT LEVEL INDICATOR	
75	0110	0261	1 011 000 111		BRN	LJSB2	NO CARRY; O.K. GO TO COMMON PART WITH JSB	
76	0111	0163	0 111 001 111		BRN	LEROR	CARRY; SYNTAX ERROR	
77	0112		0 001 011 000	LOXXB	LDC1		LOAD SYMBOL CODE FOR 1/X	
78	0113	0354	1 110 110 101		JSB	LPRINT	PRINT AND RESTORE DATA IN C-REG	
79	0114		1 100 001 100		PT12		SET POINTER	
80	0115		1 011 101 110	LOXXC	W,ZTA		CLEAR A-REG	
81	0116		1 111 100 010		P,AP1A		GENERATE "ONE" AS DIVIDEND	
82	0117		1 011 000 100	LOXXD	SS11		SET SINGLE OPERAND OP. FLAG	
83	0120	0137	0 101 111 111		BRN	LDIVR	GO TO DIVISION ROUTINE	
84	0121		0 101 011 000	LTWWB	LDC5		LOAD SYMBOL CODE FOR X/12	
85	0122	0354	1 110 110 101		JSB	LPRINT	PRINT AND RESTORE DATA IN C-REG	
86	0123		1 100 001 100		PT12			
87	0124		0 110 001 110		W,CTA		SEND NEW DATA TO A-REG (DIVISOR)	
88	0125		0 011 001 110		W,ZTC		RESET C-REG TO ZERO	
89	0126		0 001 011 000		LDC1		BUILD "12" FOR DIVISOR	
90	0127		0 010 011 000		LDC2			
91	0130		1 100 001 100		PT12		SET PROPER POINTER VALUE	
92	0131	0074	0 011 110 011		BRN	LTWW2		
93	0132	0026	0 001 011 011	LPCHS	BRN	LCHS1	GO TO CH.S.ROUTINE EXECUTION	
94	0133	0007	0 000 100 001	LLNB	JSB	LPREP	PREPARE DATA FOR PRINT	
95	0134		0 111 011 000		LDC7		LOAD SYMBOL CODE FOR LN	
96	0135	0354	1 110 110 101		JSB	LPRINT	PRINT AND RESTORE DATA	
97	0136	0001	0 000 000 111		BRN	LLN1	GO TO EXECUTION PART	
98	0137		0 110 100 100	LDIVR	RS6		RESET D.P. FLAG IF SET	
99	0140		1 000 010 000	LDIVQ	ROM 4		GO TO DIVISION ROUTINE	
100	0141	0055	0 010 110 111	LGRND	BRN	LGT1	GO TO GRAND-TOTAL ROUTINE	
101	0142		0 010 101 000	LIOC2	CXM		GET INT. FLG. VECTOR FOR UPDATE	
102	0143		1 011 001 100		PT11		PREPARE DIGIT INTERPRETATION CODE FLAG	
103	0144		0 010 011 000		LDC2		SET IT TO TWO	
104	0145	0035	0 001 110 111		BRN	LIOC3	GO TO STORE INT.FLG.VECTOR	
105	0146		0 110 011 000	LEXPB	LDC6		LOAD SYMBOL CODE FOR EXP	
106	0147	0354	1 110 110 101		JSB	LPRINT	GO TO PRINT AND RESTORE DATA IN C-REG	
107	0150	0002	0 000 001 011		BRN	LEXP1	GO TO EXECUTION PART OF THE ROUTINE	
108	0151		0 010 001 110	LINCL	W,BTC		RESTORE DATA	
109	0152		0 111 000 000	LINCR	PINC		PROGRAM COUNTER INCREMENT BY ONE	
110	0153		0 111 000 000		PINC		PROGRAM COUNTER INCREMENT BY TWO	
111	0154	0275	1 011 110 111		BRN	LFIN1	GO TO RESET FLAGS AND ROUTINE EXIT	
112	0155		0 100 101 000	LALF2	CTS		STORE RIGHT PART OF MESSAGE IN STACK	

MAIN UNIT LISTING (ROM A—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156		0 001 001 100		PTI			PREPARE COUNTER
114	0157	0174	0 111 110 011		BRN	LALF4		
115	0160		1 000 010 000	LCLA9	ROM 4			
116	0161		0 111 000 000	LCLAZ	PINC			INCREMENT PROGRAM COUNTER
117	0162	0151	0 110 100 111		BRN	LINCL		
118	0163		1 110 010 000	LEROR	ROM 7			GO TO MESSAGE ROUTINE
119	0164		0 111 000 000	LCLAN	PINC			PREPARE NEXT STEP IN SEARCH ROUTINE
120	0165		0 111 000 000	LCLAD	PINC			INCR. P-REG. TO REACH THIRD NEXT ADDR.
121	0166		0 011 001 110		W,ZTC			PREPARE C-REG
122	0167		0 101 001 100		PT5			PREPARE COUNTER
123	0170	0301	1 100 001 001		JSB	LGNTI		GO GET NEXT INSTRUCTIONS INTO T-REG
124	0171		1 110 101 110		W,AXC			GET MASTER INTO C-REG
125	0172		0 110 001 110		W,CTA			COPY IT BACK INTO A-REG FOR NEXT STEP
126	0173	0160	0 111 000 011		BRN	LCLA9		
127	0174	0303	1 100 010 001	LALF4	JSB	LGNT2		GET SEVEN INSTR. TO BUILD LEFT PART OF MESS. IN T-REG
128	0175		1 100 101 000		DNR			RIGHT PART OF THE MESSAGE INTO C-REG
129	0176		0 111 000 000	LALF3	PINC			PRGM. COUNTER INCREMENT (+10) (PT=8)
130	0177		0 000 111 100		PLS			COUNT INCREMENTS
131	0200		0 011 101 100		YP3			11-TH?
132	0201	0176	0 111 111 011		BRN	LALF3		NO; CONTINUE
133	0202	0031	0 001 100 111		BRN	LALF8		GO TO PRINT MESSAGE
134	0203		1 111 001 100	LRET2	PT15			PREPARE FOUR SHIFTS DOWN
135	0204		1 001 001 110	LALIG	W,SRC			ALIGNMENT SHIFT
136	0205		0 000 111 100		PLS			COUNT
137	0206		0 011 101 100		YP3			FINISHED?
138	0207	0204	1 000 010 011		BRN	LALIG		NO; CONTINUE
139	0210		0 101 010 100		YS5			YES,WHO?
140	0211	0214	1 000 110 011		BRN	LRET20		IT IS RETURN ROUTINE
141	0212		0 111 110 000		CTT			COPY SHIFTED ADDR. INTO T-REG
142	0213		0 000 110 000		RETURN			END OF THE SUBROUTINE
143	0214		1 011 110 000	LRET20	DTDS			STORE NEW STACK IN D.S.
144	0215		1 000 000 000		TTP			RETURN ADDR INTI PRGM.COUNTER
145	0216	0151	0 110 100 111		BRN	LINCL		
146	0217		0 001 101 110	LCLAF	W,CM1			MATCH FOUND?
147	0220	0164	0 111 010 011		BRN	LCLAN		NO; GO TO NEXT STEP
148	0221	0161	0 111 000 111		BRN	LCLAZ		YES; FINISHED
149	0222		0 010 101 000	LJMP2	CXM			STORE INT. FLG. VECTOR RESTORE DATA
150	0223		1 000 101 110		W,BXC			SAVE DATA INTO B-REG
151	0224		0 100 010 100		YS4			IS THIS JUMP RELATIVE?
152	0225	0244	1 010 010 011		BRN	LJMP3		NO;
153	0226	0301	1 100 001 001	LJAB1	JSB	LGNTI		YES; GET NEXT TWO INSTRUCTIONS (PT=6)
154	0227	0203	1 000 010 001		JSB	LALIG		ADDRESS MODIFIER ALIGNMENT
155	0230		0 100 010 000	LJAB2	ROM 2			GO TO MODIFY ADDRESS
156	0231		0 010 101 000	LJMPA	CXM			GET INT. FLG. VECTOR, SAVE DATA
157	0232		0 110 001 100		PT6			PREPARE PROGRAM JUMP CONDITION FLAG
158	0233		0 110 100 010		P,ZMC			TEST CONDITION FOR PROGRAM JUMP
159	0234	0222	1 001 001 011		BRN	LJMP2		NO CARRY; ZERO; CONDITION MET; GO TO EXECUTE
160	0235		0 000 011 000		LDC0			CARRY; CONDITION NOT MET; RESET CONDITION FLAG
161	0236		0 010 101 000		CXM			STORE INT. FLG. VECTOR; RESTORE DATA
162	0237		0 101 010 100		YS5			IS IT RETURN ROUTINE?
163	0240	0275	1 011 110 111		BRN	LFIN1		YES; SKIP PRGM. COUNTER INCREMENT
164	0241		1 001 010 100		YS9			IS IT JUMP TO C-REG CONTENTS?
165	0242	0152	0 110 101 011		BRN	LINCR		NO. GO TO PRGM COUNTER INCREMENT
166	0243	0275	1 011 110 111		BRN	LFIN1		GO TO ROUTINE EXIT
167	0244		0 001 010 100	LJMP3	YS1			IS THIS A "JSB" ROUTINE
168	0245	0320	1 101 000 011		BRN	LJMP4		NO; IT IS "GO TO" ROUTINE
169	0246		1 010 101 000		MTC			YES; GET INT. FLG. VECTOR
170	0247		0 101 001 100		PT5			PREPARE SUBROUTINE LEVEL INDICATOR
171	0250		0 101 010 100		YS5			IS THIS RETURN ROUTINE?
172	0251	0107	0 100 011 111		BRN	LRET1		YES

MAIN UNIT LISTING (ROM A—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		1 110 100 010		P,AXC		NO; IT IS JSB; PREPARE LEVEL TEST
174	0253		0 011 011 000		LDC3		SET LIMIT
175	0254		0 000 111 100		PLS		RESTORE POINTER
176	0255		0 101 000 010		P,AMCC		CHECK DIFFERENCE
177	0256	0163	0 111 001 111		BRN	LEROR	NO CARRY; LIMIT; SYNTAX ERROR
178	0257		1 110 100 010		P,AXC		CARRY, O.K., RESTORE LEVEL INDICATOR IN C-REG
179	0260		0 111 100 010		P,CP1C		INCREMENT LEVEL INDICATOR
180	0261		0 010 101 000	LJSB2	CXM		STORE UPDATED INT. FLG. VECTOR
181	0262		0 011 001 110		W,ZTC		START TO BUILD INT. D.S.R. ADDRESS
182	0263		0 111 101 110		W,CP1C		SET EXPONENT TO ONE
183	0264		1 100 001 100		PT12		PREPARE PROPER POINTER
184	0265		1 111 011 000		LDC15		INTERNAL D.S. ADDRESS
185	0266	0337	1 101 111 111		BRN	LJSB20	
186	0267		0 010 010 000	LFPRINN	ROM 1		
187	0270	0014	0 000 110 011	LOXXT	BRN	LOXXA	GO TO 1/X ROUTINE BEGINNING
188	0271	0040	0 010 000 011	LEXPT	BRN	LEXPA	GO TO EXP. ROUTINE BEGINNING
189	0272		0 000 010 000	LTRB2	ROM 0		
190	0273		0 110 010 000	LCLA2	ROM 3		GO TO SEARCH FOR LABEL ROUTINE
191	0274		0 010 001 110	LFIN4	W,BTC		RESTORE DATA IN C-REG
192	0275		0 010 010 000	LFIN1	ROM 1		ROUTINE EXIT
193	0276		0 011 001 110	LPLA1	W,ZTC		PREPARE C-REG FOR LABEL
194	0277		0 110 001 100		PT6		PREPARE COUNTER
195	0300	0301	1 100 001 001		JSB	LGNTI	GET LABEL NUMBER FROM THE PROGRAM
196	0301	0273	1 011 101 111		BRN	LCLA2	GO TO SEARCH FOR LABEL ROUTINE
197	0302		0 111 000 000	LGNTI	PINC		INCREMENT PRGM. COUNTER
198	0303		0 111 000 000		PINC		INCREMENT PRGM. COUNTER
199	0304		1 001 001 110	LGNT2	W,SRC		ALIGNMENT SHIFT ONE
200	0305		1 001 001 110		W,SRC		ALIGNMENT SHIFT TWO
201	0306		0 111 110 000		CTT		COPY C-REG TO T-REG
202	0307		1 000 100 000		READ		GET NEXT INSTRUCTION
203	0310		1 011 010 100		YS11		EXTERNAL FLAG?
204	0311	0364	1 111 010 011		BRN	LOKY2	NO; O.K.
205	0312	0272	1 011 101 011		BRN	LTRB2	YES; TROUBLE
206	0313		0 101 000 000	LJSB4	IS2		
207	0314		1 000 101 000		PTT		ADD NEW RETURN VECTOR TO STACK
208	0315		1 001 000 000		IS1		
209	0316		0 111 111 000		TCC		PREPARE NEW RET. V. STACK FOR STORE
210	0317		1 011 110 000		DTDS		STORE NEW STACK IN D.S.
211	0320		1 000 010 100	LJMP4	YS8		IS THIS JUMP TO ABS. ADDRESS?
212	0321	0226	1 001 011 011		BRN	LJAB1	YES
213	0322		1 001 010 100		YS9		NO; IT IS JUMP TO LABEL; WHERE?
214	0323	0276	1 011 111 011		BRN	LPLA1	LABEL IS IN PROGRAM
215	0324		1 100 101 110		W,AXB		LABEL IS IN CPU, GET IT INTO A-REG
216	0325		0 010 010 000		ROM 1		
217	0326		1 001 010 100	LJSB40	YS9		LABEL IN C-REG?
218	0327	0313	1 100 101 111		BRN	LJSB4	NO
219	0330		0 101 000 000		IS2		YES
220	0331		1 001 001 000		PDEC		DECREMENT CURRENT ADDRESS
221	0332		1 001 001 000		PDEC		BY TWO FOR PROPER RETURN
222	0333		1 001 000 000		IS1		
223	0334	0313	1 100 101 111		BRN	LJSB4	
224	0335	0274	1 011 110 011	LRETX	BRN	LFIN4	GO TO ROUTINE EXIT
225	0336	0273	1 011 101 111	LCLAB	BRN	LCLA2	
226	0337		0 110 011 000	LJSB20	LDC6		RETURN VECTOR STACK ADDRESS
227	0340		1 001 110 000		ATDS		D.S. CALL
228	0341		0 110 001 100		PT6		PREPARE PROPER POINTER
229	0342		1 011 111 000		DSTC		READ RETURN VECTOR STACK INTO C-REG
230	0343		0 111 110 000		CTT		IS THIS RETURN ROUTINE?
231	0344		0 101 010 100		YS5		YES
232	0345	0203	1 000 001 111		BRN	LRET2	NO; IT IS JSB, PREPARE STACK-UP (NON BCD-CH.)

MAIN UNIT LISTING (ROM A—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 000 111 100	LJSB3	PLS		COUNT SHIFTS (BEGINNING PT=10)
234	0347		0 101 000 000		IS2		
235	0350		0 100 001 000		SLT		SHIFT T-REG LEFT ONE BIT
236	0351		1 001 000 000		IS1		
237	0352		0 110 101 100		YP6		WAS IT 16-TH SHIFT?
238	0353	0346	1 110 011 011		BRN	LJSB3	NO; CONTINUE
239	0354	0326	1 101 011 011		BRN	LJSB40	YES, FINISHED
240	0355		0 101 010 100	LPRINT	YS5		PRINT OR NOT TO PRINT?
241	0356	0267	1 011 011 111		BRN	LFPRINN	GO TO PRINT
242	0357		0 101 100 100		RS5		
243	0360		0 010 001 110		W,BTC		
244	0361		0 000 110 000		RETURN		
245	0362		0 010 100 000		RMGRB		CALL F. BLOCK FOR INTERPRETATION
246	0363	0050	0 010 100 011	LTWWT	BRN	LTWWA	GO TO X/12 ROUTINE BEGINNING
247	0364		0 101 000 000	LOKY2	IS2		
248	0365		0 100 101 000		IXT		LOAD NEW INSTRUCTION INTO T-REG
249	0366		1 001 001 000		PDEC		DECREMENT PROGRAM COUNTER
250	0367		1 001 000 000		IS1		
251	0370		0 111 111 000		TTC		COPY NEW INSTRUCTION INTO C-REG
252	0371		0 000 111 100		PLS		COUNT
253	0372		1 000 101 100		YP8		FINISHED?
254	0373	0304	1 100 010 011		BRN	LGNT2	NO; MAKE THE LOOP AGAIN
255	0374		0 000 110 000		RETURN		YES; SUBROUTINE END
256	0375	0101	0 100 000 111	LLOGT	BRN	LLOGB	GO TO LOG ROUTINE BEGINNING
257	0376	0133	0 101 101 111	LLNT	BRN	LLNB	GO TO LN ROUTINE BEGINNING
258	0377	0142	0 110 001 011	LIOPC	BRN	LIOC2	GO TO I/O CALL ROUTINE

MAIN UNIT LISTING (ROM A—7)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000	0014	0 000 110 011		BRN	LSYNG
4	0001		1 101 111 110	LADJ2	S,AM1A	RESTORE D.P.L. FLAG
5	0002		1 101 111 110	LADJ4	S,AM1A	DECREMENT, CHECK IF CARRY
6	0003	0010	0 000 100 011		BRN	LADJ3 NO CARRY CONTINUE EXPONENT ADJUSTMENT
7	0004		0 001 001 100		PT1	CARRY, ADJUSTMENT OF THE EXP. IS FINISHED
8	0005		1 001 100 010		P,AM1	CHECK MOST SIGNIFICANT DIGIT OF THE EXPONENT
9	0006	0322	1 101 001 011		BRN	LTRB1
10	0007	0314	1 100 110 011		BRN	LTRB3 ZERO,GO TO SIGN OF EXP.CHECK
11	0010		1 111 101 010	LADJ3	X,AP1A	INCREMENT EXPONENT
12	0011		0 000 000 000		NOP	TO ELIMINATE CARRY
13	0012	0002	0 000 001 011		BRN	LADJ4
14	0013		1 010 001 100	LSYNA	PT10	SET POINTER FOR NOTE 05
15	0014		0 000 110 100	LSYNG	CLS	CLEAR ALL STATUS BITS
16	0015	0237	1 001 111 111		BRN	LSYNN
17	0016		1 001 000 100	LGT3	SS9	SET SHIFT FLAG
18	0017		0 000 010 000		ROM 0	
19	0020		1 000 101 110	LERRR	W,BXC	
20	0021	0221	1 001 000 111		BRN	LERA0
21	0022		1 011 011 000	LGTS	LDC11	R (FROM ROM 6)
22	0023		1 011 011 000		LDC11	LOAD T
23	0024	0231	1 001 100 111		BRN	LGT4
24	0025	0165	0 111 010 111	LPREPA	BRN	LPRSA
25	0026		0 000 010 000	LRES5	ROM 0	RETURN TO ROM 0
26	0027		0 010 010 000	LFPR1	ROM 1	GO TO PRINT PART
27	0030	0365	1 111 010 111	LRESB	BRN	LRES7
28	0031		0 010 101 000	LRES8	CXM	PUT INT.FLG.VECTOR INTO M-REG
29	0032		1 101 001 100		PT13	PREPARE MESSAGE GEN.
30	0033		1 111 011 000	LRES4	LDC15	"BLANK"
31	0034		0 001 101 100		YP1	BLANKS STILL NEEDED?
32	0035	0033	0 001 101 111		BRN	LRES4 YES
33	0036		0 001 011 000		LDC1	NO, SET "C"
34	0037		0 111 011 000		LDC7	SET "L"
35	0040		0 111 110 000		CTT	STORE LEFT PART OF THE PRNT. WORD IN T-REG
36	0041		1 101 001 100		PT13	
37	0042		1 011 011 000		LDC11	SET "E"
38	0043		0 000 000 000		NOP	
39	0044		1 100 011 000		LDC12	SET "A"
40	0045		1 011 011 000		LDC11	SET "R"
41	0046		1 111 011 000		LDC15	SET "BLANK"
42	0047	0026	0 001 011 011		BRN	LRES5 END OF MESSAGE
43	0050		1 111 011 000	CONT	LDC15	LOAD BLANKS
44	0051		0 100 101 000		CTS	STORE PRINT MASK
45	0052		1 101 001 100		PT13	
46	0053		1 111 011 000	LMESA	LDC15	
47	0054		0 111 101 100		YP7	
48	0055	0053	0 010 101 111		BRN	LMESA
49	0056		1 100 011 000		LDC12	N
50	0057		1 111 011 000		LDC15	BLANK
51	0060		0 000 011 000		LDC0	O
52	0061		1 111 011 000		LDC15	BLANK

MAIN UNIT LISTING (ROM A—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
53	0062		1 011 011 000		LDC11	T
54	0063		1 111 011 000		LDC15	BLANK
55	0064		1 011 011 000		LDC11	E
56	0065		1 111 011 000		LDC15	
57	0066		0 111 110 000		CTT	
58	0067		1 100 101 000		DNR	RECOVER PRINT MASK
59	0070		0 000 110 000		RETURN	END OF THE SUBROUTINE
60	0071		0 110 010 000	LPRS6	ROM 3	
61	0072		1 011 111 000		DSTC	GET GRAND TOTAL
62	0073	0016	0 000 111 011		BRN LGT3	
63	0074		0 100 101 110	LRED1	W,ATB	COPY ORIG. DATA INTO B-REG
64	0075		0 000 011 110		S,ZMB	CHECK SIGN OF MANTISSA
65	0076	0100	0 100 000 011		BRN LADJ1	POSITIVE, GO TO DATA ADJUSTMENT
66	0077		0 001 110 000		RED	PREPARE RED TAPE
67	0100		1 110 110 110	LADJ1	MS,AXC	MANTISSA BACK TO C-REG; FLG. VECTOR TO a-REG
68	0101		0 011 011 010		XS,ZTC	CANCELL OP.CODE IN C-REG
69	0102		1 111 111 110		S,AP1A	INCREMENT; CHECK IF CARRY
70	0103	0001	0 000 000 111		BRN LADJ2	NO, CONTINUE TEST
71	0104		0 110 001 110	LSCI1	W,CTA	YES; START SCI. NOTATION FORMAT PRINT
72	0105		1 011 001 100		PT11	SET POINTER TO INDICATE DE.POINT LOCATION
73	0106		0 101 000 100		SS5	SET SCI. NOT. FORMAT FLAG
74	0107		1 011 010 100	LDCP1	YS11	CHECK IF EXTERNAL FLAG IS SET
75	0110	0133	0 101 101 111		BRN LDCP4	NO, CONTINUE IN PRINT ROUTINE
76	0111		0 100 010 000		ROM 2	YES!GO TO DISPLAY ROUTINE
77	0112		1 001 100 100	LFPRM	RS9	
78	0113		0 010 001 110		W,BTC	RESTORE DATA IN C-REG
79	0114		0 000 110 000		RETURN	
80	0115		1 111 101 010	LTRB4	X,AP1A	
81	0116	0256	1 010 111 011		BRN LTRB5	
82	0117		0 011 011 110	LADJ6	S,ZTC	PREPARE ALIGNMENT SHIFT
83	0120		1 001 001 110		W,SRC	MANTISSA ALIGNMENT IN FIX FORMAT PRINT
84	0121		0 000 001 100	LADJ5	PT0	PREPARE CHECK OF THE LEAST SIGN DIGIT
85	0122		1 111 100 010		P,AP1A	INCREMENT TO CHECK IF IT WAS NINE
86	0123	0117	0 100 111 111		BRN LADJ6	NO CARRY, GO TO ROUND AND SHIFT
87	0124	0261	1 011 000 111		BRN LADJ7	GO TO ROUND OFF ROUTINE
88	0125		1 110 101 110	LADJ8	W,AXC	DATA TEMP. INTO A-REG
89	0126		1 010 101 000		MTC	PREPARE INT.FLG.VECTOR FOR TEST
90	0127		0 010 001 100		PT2	PRESET POINTER FOR FIX 0 FORMAT
91	0130		0 101 111 110	LDCP3	S,CM1C	DECREMENT DPL FLAG TO CHECK IF IT IS ZERO
92	0131	0150	0 110 100 011		BRN LDCP2	NOT ZERO, CONTINUE
93	0132	0107	0 100 011 111		BRN LDCP1	ZERO, FINISHED
94	0133		1 110 101 110	LDCP4	W,AXC	BRING PART OF PRINT, WORD TO C-REG
95	0134		1 001 010 010		WP,SRC	MAKE SPACE FOR DEC.POINT PRINT CODE
96	0135		0 111 100 010		P,CP1C	SET "1" TO INDICATE D.P. LOCATION
97	0136		0 110 001 110		W,CTA	COPY NEW "MASK" TO A-REG
98	0137		1 010 011 000		LDC10	SET PRINT CODE FOR D.P.
99	0140		1 100 001 100		PT12	PREPARE SEARCH FOR LEADING ZEROS
100	0141		1 001 100 010	LDCP5	P,AM1	DECREMENT, CHECK IF CARRY
101	0142	0347	1 110 011 111		BRN LMSS1	NON ZERO, FINISHED
102	0143		1 111 011 000		LDC15	ZERO, SET CODE FOR BLANK INTO C-REG
103	0144	0141	0 110 000 111		BRN LDCP5	CONTINUE IN SEARCH
104	0145		1 110 101 110	LSYM3	W,AXC	COPY EXPONENT INTO C-REG
105	0146		1 111 011 000	LSYM9	LDC15	LOAD PRINT CODE FOR BLANK
106	0147	0071	0 011 100 111		BRN LPRS6	GO TO SUBROUTINE EXIT
107	0150		0 000 111 100	LDCP2	PLS	INCREMENT POINTER TO INDICATE FIX FORMAT
108	0151	0130	0 101 100 011		BRN LDCP3	
109	0152		0 101 000 100	LPRS5	SS5	SET "DO NOT PRINT" FLAG
110	0153		0 100 101 110		W,ATB	COPY DATA INTO B-REG FOR NEXT EXECUTION STEPS
111	0154	0071	0 011 100 111		BRN LPRS6	
112	0155		1 110 100 000	LPRS4	YPOC	CHECK PRINT-ON-COMMAND FLAG

MAIN UNIT LISTING (ROM A—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	0156		1 011 010 100	YS11		ON COMMAND ONLY?
114	0157	0242	1 010 001 011	BRN	LREDQ	GO TO CHECK AUX.FLAG
115	0160		1 011 100 100	RS11		RESET EXTERNAL FLAG
116	0161		0 101 010 100	LPRS50 YS5		IS AUX.FLAG SET?
117	0162	0152	0 110 101 011	BRN	LPRS5	NO,CONTINUE IN DATA ADJUSMENR
118	0163	0276	1 011 111 011	BRN	LFPRS	YES,GO TO LEFT PARENTHESIS ROUTINE
119	0164	0014	0 000 110 011	LERROR BRN	LSYNG	
120	0165		0 110 001 110	LPRSA W,CTA		COPY DATA TO A-REGISTER
121	0166		1 010 101 000	LPRSAA MTC		GET INT.FLG.VECTOR FOR TEST
122	0167		1 100 001 100	PT12		PREPARE OP. MODE CODE FOR TEST
123	0170		0 110 100 010	P,ZMC		CHECK IF ZERO
124	0171	0155	0 110 110 111	BRN	LPRS4	OP. CODE WAS NOT ONE, CONTINUE TEST
125	0172	0161	0 111 000 111	BRN	LPRS50	GO TO CHECK AUX.FLAG
126	0173		0 000 110 100	LNOT20 CLS		CLEAR ALL STATUS BITS TO AVOID TROUBLE
127	0174	0176	0 111 111 011	BRN	NOT20	
128	0175	0263	1 011 010 001	LERIA JSB	LMESS	GENERATE MESSAGE
129	0176	0324	1 101 010 101	NOT20 JSB	LFPRZ	PRINT
130	0177	0247	1 010 011 111	BRN	LSYN9	EXIT
131	0200		0 101 010 100	LPRTA YS5		PRINT OR NOT?
132	0201	0027	0 001 011 111	BRN	LFPR1	PRINT
133	0202		1 010 101 000	LFPR4 MTC		DO NOT; GET INT. FLAG VECTOR FOR UPDATE
134	0203		0 001 001 100	PT1		PREPARE OP. CODE LOADING
135	0204		0 011 010 010	WP,ZTC		MASK PREVIOUS OP. CODE WITH ZEROS
136	0205		0 101 100 100	RS5		RESET "DO NOT PRINT" FLAG IF SET
137	0206	0220	1 001 000 011	BRN	LFPR5	
138	0207		0 111 110 000	LSCI3 CTT		STORE LEFT HALF OF THE PRINT WORD
139	0210		1 100 101 010	X,AXB		SEND EXPONENT TO A-REG
140	0211		0 100 101 010	X,ATB		RESTORE ORIG. EXPONENT IN B-REG
141	0212		0 011 001 110	LSCI4 W,ZTC		CANCELL NONBCD CHAR. IN C-REG
142	0213		0 100 001 110	LSCI5 W,SLA		ALIGNMENT SHIFT
143	0214		0 000 111 100	PLS		INCREMENT SHIFT COUNTER
144	0215		1 011 101 100	YP11		FINISHED?
145	0216	0213	1 000 101 111	BRN	LSCI5	NO, STAY IN LOOP
146	0217	0145	0 110 010 111	BRN	LSYM3	YES
147	0220		0 000 010 000	LFPR5 ROM 0		GO BACK TO ROM 0
148	0221	0263	1 011 010 001	LERA0 JSB	LMESS	GO TO MESSAGE GEN.
149	0222		1 101 001 100	PT13		LOAD 20
150	0223	0373	1 111 101 111	BRN	NOT2	
151	0224		0 110 001 110	LPRXA W,CTA		GO TO FORCED PRINT DATA ADJUSTMENT
152	0225		1 010 101 000	MTC		
153	0226	0074	0 011 110 011	BRN	LRED1	
154	0227		0 100 101 000	LPRST CTS		STORE SYMBOL TEMPORARILY IN D-REG
155	0230	0166	0 111 011 011	BRN	LPRSAA	
156	0231		0 000 010 000	LGT4 ROM 0		GO TO PRINT
157	0232		1 011 111 000	LAC1 DSTC		FETCH ACCUMULATED TOTAL
158	0233		0 000 110 100	CLS		CLEAR STATUS BITS
159	0234		1 010 000 100	SS10		SET EQUAL KEY FLAG
160	0235		0 110 000 100	SS6		SET L-FLAG
161	0236	0317	1 100 111 111	BRN	LAC2	
162	0237		0 111 000 100	LSYNN	SS7	
163	0240		0 010 100 100	LSYNF RS2		
164	0241	0175	0 111 110 111	BRN	LERIA	
165	0242		0 101 010 100	LREDQ YS5		IS AUX.FLAG SET?
166	0243	0074	0 011 110 011	BRN	LRED1	NO, GO TO CHECK SIGN
167	0244		1 100 101 000	DNR		RESTORE CODED SYMBOL FOR PRINT IN C-REG
168	0245	0027	0 001 011 111	BRN	LFPR1	YES,GO TO FORCED PRINT ROUTINE
169	0246	0221	1 001 000 111	BRN	LERA0	
170	0247		1 100 001 100	LSYN9 PT12		SET POINTER TO CHANGE MODE
171	0250	0310	1 100 100 011	BRN	LSYN2	GO TO ROUTINE EXIT
172	0251		1 001 110 000	LAC0 ATDS		SEND ADDRESS TO DATA STORAGE

MAIN UNIT LISTING (ROM A—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252	0232	1 001 101 011		BRN	LAC1	
174	0253		1 111 111 010	LTRB2	XS,AP1A		CHECK SIGN OF EXP.
175	0254	0104	0 100 010 011		BRN	LSCI1	NOT NINE,SCI.NOT.FMT.
176	0255		1 101 111 010		XS,AM1A		NINE,RESTORE SIGN OF THE EXP.
177	0256		0 011 010 110	LTRB5	MS,ZTC		
178	0257		0 110 010 110		MS,CTA		COPY ZERO MANTISSA TO A-REG
179	0260	0121	0 101 000 111		BRN	LADJ5	CONTINUE IN ADJUSTMENT FOR FIX.P.FMT.
180	0261		0 100 010 000	LADJ7	ROM 2		
181	0262		1 001 011 000		LDC9		
182	0263	0354	1 110 110 011		BRN	LART1	
183	0264		0 011 001 110	LMESS	W,ZTC		
184	0265		0 000 011 100	UPDATE	PRS		GENERATE NOTE NUMBER
185	0266		0 111 111 110		S,CP1C		
186	0267		1 011 101 100	YP11			
187	0270	0265	1 011 010 111		BRN	UPDATE	
188	0271		1 001 001 110		W,SRC		SHIFT NOTE NUMBER
189	0272		1 011 100 100		RS11		RESET STATUS 11 IF SET
190	0273		1 111 011 000	LDC15			
191	0274	0050	0 010 100 011		BRN	CONT	
192	0275	0221	1 001 000 111		BRN	LERA0	
193	0276		0 101 010 100	LFPRS	YS5		AUX. FLAG?
194	0277	0202	1 000 001 011		BRN	LFPR4	NOT SET, FINISHED, GO TO SUBROUTINE EXIT
195	0300		0 110 010 000		ROM 3		SET, RETURN TO LEFT PAR. ROUTINE
196	0301		0 111 110 000	LFIX1	CTT		STORE LEFT HALF OF THE PRINT WORD
197	0302		1 101 001 100		PT13		SET POINTER TO SUPRESS EXPONENT PRINT
198	0303		1 111 011 000		LDC15		LOAD "BLANK" INSTEAD OF EXP. M.S.D.
199	0304		0 000 000 000		NOP		BECAUSE OF HARDWARE "FEATURE"
200	0305		0 011 010 010		WP,ZTC		SET ZERO FOR OP.SYMBOL CODE GEN.BEGINNING
201	0306		1 111 011 000		LDC15		LOAD BLANK INSTEAD OF L.S.D. OF EXP.
202	0307	0146	0 110 011 011		BRN	LSYM9	GO TO SUBROUTINE EXIT
203	0310		0 000 010 000	LSYN2	ROM 0		RETURN TO ROM 0
204	0311		0 010 001 100	LRES9	PT2		PREPARE OPCODE LOAD
205	0312		0 001 011 000		LDC1		LOAD OP.CODE FOR ADD
206	0313	0031	0 001 100 111		BRN	LRES8	
207	0314		1 001 111 010	LTRB3	XS,AM1		CHECK SIGN OF EXP.
208	0315	0253	1 010 101 111		BRN	LTRB2	NONZERO,CONTINUE IN TEST
209	0316	0121	0 101 000 111		BRN	LADJ5	ZERO,O.K. FIX.P.FORMAT
210	0317		0 011 000 000	LAC2	SS3		SET OP FLAG TO INDICATE GRANDTOTAL ACC ROUTE
211	0320		1 100 101 110		W,AXB		PREPARE DATA IN A-REG FOR SUMMATION
212	0321		1 000 010 000		ROM 4		DO AN ACC +
213	0322		1 001 111 010	LTRB1	XS,AM1		
214	0323	0115	0 100 110 111		BRN	LTRB4	
215	0324	0104	0 100 010 011		BRN	LSCI1	
216	0325		1 001 000 100	LFPRZ	SS9		SET FORCED PRINT FLAG
217	0326	0027	0 001 011 111		BRN	LFPR1	
218	0327		1 111 011 000	LMSS2	LDC15		LOAD PRINT CODE FOR BLANK (PLUS)
219	0330		0 001 001 100	LMSS3	PT1		PREPARE EXPONENT PRINT CANCELLATION
220	0331		1 111 011 000		LDC15		LOAD "BLANK"
221	0332		1 111 011 000		LDC15		LOAD "BLANK"
222	0333		0 101 010 100		YS5		CHECK FORMAT FLAG
223	0334	0301	1 100 000 111		BRN	LFIX1	FIX. POINT
224	0335		0 101 100 100		RS5		SCI. NOT. RESET FLAG
225	0336		0 000 011 010		XS,ZMB		CHECK EXP. SIGN OF ORIGINAL DATA
226	0337	0207	1 000 011 111		BRN	LSCI3	POSITIVE
227	0340		0 000 001 100		PT0		NEGATIVE
228	0341		1 001 011 000		LDC9		LOAD CODE FOR MINUS
229	0342		0 111 110 000		CTT		STORE LEFT HALF OF THE PRINT WORD
230	0343		0 010 001 010		X,BTC		LOAD ORIG. EXPONENT INTO C-REG
231	0344		0 010 101 010		X,ZMCC		COMPLEMENT OF THE EXPONENT
232	0345		0 110 001 010		X,CTA		SEND EXPONENT INTO A-REG

MAIN UNIT LISTING (ROM A—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
233	0346	0212	1	000	101	011		BRN	LSCI4	GO TO EXPONENT ALIGNMENT BY SHIFTING
234	0347		1	101	001	100	LMSS1	PT13		PREPARE OPERATION ON SIGN OF MANTISSA
235	0350		0	000	011	110		S,ZMB		TEST ORIG. SIGN OF MANTISSA
236	0351	0327	1	101	011	111		BRN	LMSS2	POSITIVE
237	0352		1	001	011	000		LDC9		NEGATIVE, LOAD PRINT CODE FOR MINUS
238	0353	0330	1	101	100	011		BRN	LMSS3	
239	0354		0	011	010	010	LART1	WP,ZTC		
240	0355		0	101	110	010		WP,CM1C		
241	0356		0	011	011	010	LART2	XS,ZTC		
242	0357		1	000	101	110		W,BXC		
243	0360	0263	1	011	010	001		JSB	LMESS	
244	0361		0	111	111	110		S,CP1C		
245	0362	0375	1	111	110	111		BRN	LART4	
246	0363	0356	1	110	111	011		BRN	LART2	
247	0364	0354	1	110	110	011		BRN	LART1	
248	0365		0	111	111	000	LRES7	TTC		FETCH INT. FLAG VECTOR OUT OF T REG
249	0366	0311	1	100	100	111		BRN	LRES9	
250	0367		1	001	010	100	LFPRK	YS9		CHECK IF FORCED PRINT FLAG IS SET
251	0370	0276	1	011	111	011		BRN	LFPRS	NO
252	0371	0112	0	100	101	011		BRN	LFPRM	YES
253	0372	0240	1	010	000	011		BRN	LSYNF	
254	0373		0	010	011	000	NOT2	LDC2		
255	0374	0173	0	111	101	111		BRN	LNOT20	
256	0375	0324	1	101	010	101	LART4	JSB	LFPRZ	
257	0376		0	100	010	000	LOUT2	ROM 2		
258	0377	0125	0	101	010	111		BRN	LADJ8	

DATA STORAGE UNIT LISTING (ROM C—Ø)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
3	0000	0076	0 011 111 101	CACPR	JSB	CPRIT	PRINT SYMBOL	
4	0001	0315	1 100 110 111		BRN	CSUPE9	GO TO RESTORE DATA IN C-REG AND SUPERVISOR	
5	0002		0 010 010 000	CLIS1	ROM 1			
6	0003		0 011 010 000	CSTO6	TKRA		ACCEPT KEYCODE	
7	0004		1 010 101 000	CRESZ	MTC		PREPARE INT. FLG. VECTOR	
8	0005		0 000 100 000		RMGRA		GO TO RESET ROUTINE	
9	0006	0134	0 101 110 101	CSTD2	JSB	CCODE	GO TO LOAD OP. CODE	
10	0007		0 010 101 000	CSTD3	CXM		RESTORE C- AND M-REGISTERS	
11	0010	0106	0 100 011 011		BRN	CSUPER	GO TO MAIN UNIT SUPERVISOR	
12	0011		0 000 000 000		DUMMY			
13	0012		0 100 010 000	CSTD1	ROM 2		STORE-DIVIDE BEGINNING	
14	0013		0 100 010 000	CSTM1	ROM 2		STORE-MULTIPLY BEGINNING	
15	0014	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
16	0015		0 000 000 000		DUMMY			
17	0016	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
18	0017	0340	1 110 000 011	CEXCH	BRN	CEXC1	EXCH() ROUTINE BEGINNING POINT	
19	0020	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
20	0021		0 000 000 000		DUMMY			
21	0022	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
22	0023	0253	1 010 101 111	CDIG0	BRN	CDIGA	ACCEPT DIGIT Ø	
23	0024		0 000 100 000	CDIGB	RMGRA		GO TO MAIN UNIT DIGIT ENTRY	
24	0025		0 000 100 000	CLPR1	RMGRA		GO TO LPAR ROUTINE ON MAIN UNIT	
25	0026	0025	0 001 010 111	CLPAR	BRN	CLPR1	GO TO MAIN UNIT SOFTWARE	
26	0027		0 000 000 000		DUMMY			
27	0030		0 111 011 000	CSTA2	LDC7		OP. CODE Ø7	
28	0031	0007	0 000 011 111		BRN	CSTD3	GO TO RESTORE C- AND M- REGISTERS	
29	0032		1 111 101 010	CDIG3	X,AP1A		ACCEPT DIGIT 3	
30	0033		1 111 101 010	CDIG3	X,AP1A		ACCEPT DIGIT 2	
31	0034		1 111 101 010	CDIG1	X,AP1A		ACCEPT DIGIT 1	
32	0035	0023	0 001 001 111		BRN	CDIG0		
33	0036	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
34	0037		0 010 100 000	CSTO7	RMGRB		CALL F.BLK.SOFTWARE	
35	0040		0 100 010 000	CSTA1	ROM 2		STORE-ADD BEGINNING	
36	0041	0030	0 001 100 011		BRN	CSTA2		
37	0042		1 111 101 010	CDIG6	X,AP1A		ACCEPT DIGIT 6	
38	0043		1 111 101 010	CDIG5	X,AP1A		ACCEPT DIGIT 5	
39	0044		1 111 101 010	CDIG4	X,AP1A		ACCEPT DIGIT 4	
40	0045	0032	0 001 101 011		BRN	CDIG3		
41	0046	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
42	0047	0301	1 100 000 111	CRECL	BRN	CREC1	RECALL () ROUTINE BEGINNING POINT	
43	0050		0 100 010 000	CSTS1	ROM 2		STORE-SUBJECT BEGINNING	
44	0051	0060	0 011 000 011		BRN	CSTS2		
45	0052		1 111 101 010	CDIG9	X,AP1A		ACCEPT DIGIT 9	
46	0053		1 111 101 010	CDIG8	X,AP1A		ACCEPT DIGIT 8	
47	0054		1 111 101 010	CDIG7	X,AP1A		ACCEPT DIGIT 7	
48	0055	0042	0 010 001 011		BRN	CDIG6		
49	0056	0101	0 100 000 111		BRN	CDSM2	FALSE KEY	
50	0057	0226	1 001 011 011	CSTOR	BRN	CSTO1	STORE () ROUTINE BEGINNING POINT	
51	0060		1 000 011 000	CSTS2	LDC8		OP. CODE Ø8	
52	0061	0007	0 000 011 111		BRN	CSTD3	GO TO RESTORE C- AND D- REGISTERS	

DATA STORAGE UNIT LISTING (ROM C—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	Ø062	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	FALSE KEY
54	Ø063	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	FALSE KEY
55	Ø064	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	FALSE KEY
56	Ø065	ØØØ2	Ø ØØØ ØØ1 Ø11	CLIST	BRN	CLIS1	LIST AND RESET D.S. BEGINNING
57	Ø066		1 Ø1Ø 1ØØ 1ØØ	CANEN	RS1Ø		RESET SHIFT FLAG IF SET D.S. CANCEL ENTRY
58	Ø067	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	GO TO WAIT FOR NEXT KEY
59	Ø07Ø		1 Ø1Ø ØØØ 1ØØ	CSHIFT	SS1Ø		SET SHIFT FLAG
6Ø	Ø071	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	GO TO WAIT FOR NEXT KEY
61	Ø072	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	FALSE KEY
62	Ø073		Ø 1ØØ Ø1Ø ØØØ	CRESØ	ROM 2		GO TO ROM 2 FOR RESET ROUTINE START
63	Ø074		Ø ØØØ 1ØØ ØØØ		RMGRA		GO TO MAIN UNIT RESET ROUTINE
64	Ø075		Ø Ø1Ø 1Ø1 ØØØ	CDSM1	CXM		RESTORE C- AND M-REGISTERS
65	Ø076	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	FALSE KEY
66	Ø077		Ø Ø1Ø 1Ø1 ØØØ	CPRIT	CXM		GET INT.FLG.VECTOR FOR TEST
67	Ø1ØØ	Ø372	1 111 1Ø1 Ø11		BRN	CPRYT	
68	Ø1Ø1		Ø ØØØ 1ØØ 1ØØ	CDSM2	RSØ		RESET KEY-DOWN FLAG BEFORE TEST
69	Ø1Ø2		Ø ØØØ Ø1Ø 1ØØ		YSØ		IS THE OLD KEY STILL DOWN?
7Ø	Ø1Ø3	Ø12Ø	Ø 1Ø1 ØØØ Ø11		BRN	CSTO2	NO; CONTINUE IN STORE () ROUTINE
71	Ø1Ø4	Ø1Ø1	Ø 1ØØ ØØØ 111		BRN	CDSM2	YES; WAIT
72	Ø1Ø5		Ø 111 ØØØ 1ØØ	CSUPE1	SS7		SET FIRST DATA ENTRY DIGIT FLAG
73	Ø1Ø6		Ø ØØØ 1ØØ ØØØ	CSUPER	RMGRA		GO TO ROM GR. A SUPERVISOR
74	Ø1Ø7		1 Ø11 11Ø ØØØ	CFIN1	DTDS		STORE NEW DATA INGIVEN D.S. REGISTER
75	Ø11Ø		1 1ØØ 1Ø1 ØØØ		DNR		OLD RESULT BACK TO C-REG
76	Ø111		Ø 1ØØ 1Ø1 ØØØ		CTS		COPIED BACK TO D-REG ALSO
77	Ø112		Ø 11Ø 1ØØ 1ØØ		RS6		RESET STATUS SIX IF SET
78	Ø113	Ø1Ø5	Ø 1ØØ Ø1Ø 111		BRN	CSUPE1	GO TO SUPERVISOR
79	Ø114		Ø Ø1Ø Ø1Ø ØØØ	CCINC	ROM 1		
8Ø	Ø115		Ø ØØØ ØØØ ØØØ		DUMMY		
81	Ø116		Ø Ø1Ø Ø1Ø ØØØ	CDEC2	ROM 1		
82	Ø117	Ø1Ø7	Ø 1ØØ Ø11 111	CINDE	BRN	CFIN1	
83	Ø12Ø		Ø 1Ø1 ØØØ ØØØ	CSTO2	IS2		
84	Ø121		Ø 1ØØ 1ØØ ØØØ		RBL		RESET BUSY LIGHT
85	Ø122		1 ØØ1 ØØØ ØØØ		IS1		
86	Ø123		Ø ØØØ Ø1Ø 1ØØ	CSTO3	YSØ		NEW KEY DOWN?
87	Ø124	Ø123	Ø 1Ø1 ØØ1 111		BRN	CSTO3	NO; WAIT
88	Ø125		Ø ØØØ ØØ1 1ØØ		PTØ		
89	Ø126		1 Ø11 1ØØ Ø1Ø		P,ZTA		PREPARE A-REG TO ACCEPT DIGIT
9Ø	Ø127		Ø ØØØ Ø11 1ØØ	CSTO4	PRS		WAIT FOR 3MS.
91	Ø13Ø		1 1ØØ 1Ø1 1ØØ		YP12		FINISHED?
92	Ø131	Ø127	Ø 1Ø1 Ø11 111		BRN	CSTO4	NO; WAIT
93	Ø132		1 ØØØ 1Ø1 ØØØ		DSOF		YES; START EXECUTION
94	Ø133		1 1ØØ 1ØØ ØØØ		YFKB		INTERROGATE F.BLK. KBD.FLAG
95	Ø134	Ø221	1 ØØ1 ØØØ 111		BRN	CSTO5	
96	Ø135		Ø Ø1Ø 1Ø1 ØØØ	CCODE	CXM		
97	Ø136		Ø Ø11 ØØ1 1ØØ		PT3		
98	Ø137		Ø ØØ1 Ø11 ØØØ		LDC1		
99	Ø14Ø		Ø Ø11 Ø1Ø Ø1Ø	CCODG	WP,ZTC		
1ØØ	Ø141		Ø ØØØ 11Ø ØØØ		RETURN		
1Ø1	Ø142		Ø Ø1Ø 1Ø1 ØØØ	CCODF	CXM		
1Ø2	Ø143		Ø Ø11 ØØ1 1ØØ		PT3		
1Ø3	Ø144		Ø ØØØ Ø11 ØØØ		LDCØ		
1Ø4	Ø145	Ø14Ø	Ø 11Ø ØØØ Ø11		BRN	CCODG	
1Ø5	Ø146		Ø ØØ1 1ØØ 1ØØ	CCINØ	RS1		RESET AUX. FLAG
1Ø6	Ø147	Ø114	Ø 1ØØ 11Ø Ø11		BRN	CCINC	GO TO ROM 1 FOR EXECUTION
1Ø7	Ø15Ø		Ø ØØØ 1ØØ ØØØ	CMANA	RMGRA		GO TO MAIN UNIT SOFTWARE
1Ø8	Ø151		Ø ØØØ ØØØ ØØØ		DUMMY		
1Ø9	Ø152		Ø Ø1Ø 1Ø1 ØØØ	CMANL	CXM		GET INT. FLG. VECTOR FOR TEST
11Ø	Ø153		Ø 1ØØ ØØ1 1ØØ		PT4		PREPARE USERS CODE FOR TEST
111	Ø154		Ø ØØ1 1ØØ Ø1Ø		P,CM1		IS IT ZERO?
112	Ø155	Ø232	1 ØØ1 1Ø1 Ø11		BRN	CDSM9	NO; STAY IN GROUP C

DATA STORAGE UNIT LISTING (ROM C—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	Ø156		Ø Ø1Ø 1Ø1 ØØØ		CXM		YEST; RESTORE C- AND M- REGISTERS
114	Ø157	Ø15Ø	Ø 11Ø 1ØØ Ø11		BRM	CMANA	GO BACK TO MAIN UNIT SOFTWARE
115	Ø16Ø		Ø ØØØ 1ØØ ØØØ	CCOMA	RMGRA		GO TO EXECUTE OLD OPERATION
116	Ø161		Ø 1Ø1 1ØØ Ø1Ø	CDSC1	P,CM1C		IS IT ONE?
117	Ø162	Ø175	Ø 111 11Ø 111		BRN	CDSC2	NO
118	Ø163		Ø Ø1Ø Ø1Ø ØØØ		ROM 1		YES; GO TO LIST DATA ROUTINE
119	Ø164	Ø321	1 1Ø1 ØØØ 111	CCOM8	BRN	CACP4	GO TO COMMON OUTPUT TO ROMGRA
12Ø	Ø165		Ø 1Ø1 1ØØ Ø1Ø	CDSC3	P,CM1C		IS IT THREE?
121	Ø166	Ø344	1 11Ø Ø1Ø Ø11		BRN	CDSC4	NO
122	Ø167		1 ØØ1 Ø11 ØØØ		LDC9		YES; SET IT TO NINE TO INDICATE RETURN
123	Ø17Ø		Ø Ø1Ø 1Ø1 ØØØ		CXM		STORE UPDATED INT. FLG. VECTOR; RESTORE C-REG
124	Ø171		1 Ø11 ØØ1 1ØØ		PT11		LOAD SYMBOLS FOR STORE ()
125	Ø172		Ø ØØØ Ø11 ØØØ		LDCØ		S
126	Ø173		Ø Ø11 Ø11 ØØØ		LDC3		→
127	Ø174	Ø2Ø7	1 ØØØ Ø11 111		BRN	CCOMP	GO TO COMMON PRINT AND EXECUTION
128	Ø175		Ø 1Ø1 1ØØ Ø1Ø	CDSC2	P,CM1C		IS IT TWO?
129	Ø176	Ø165	Ø 111 Ø1Ø 111		BRN	CDSC3	NO
13Ø	Ø177		Ø ØØØ Ø11 ØØØ		LDCØ		YES; RESET USERS CODE
131	Ø2ØØ		Ø ØØ1 ØØ1 1ØØ		PT1		LOAD OP. CODE FOR EXCHANGE
132	Ø2Ø1		Ø ØØ1 Ø11 ØØØ		LDC1		1
133	Ø2Ø2		1 ØØØ Ø11 ØØØ		LDC8		8
134	Ø2Ø3		Ø Ø1Ø 1Ø1 ØØØ		CXM		STORE UPDATED INT. FLG. VECTOR; RESTORE C-REG
135	Ø2Ø4		1 Ø11 ØØ1 1ØØ		PT11		LOAD SYMBOLS FOR EXCHANGE
136	Ø2Ø5		Ø ØØØ Ø11 ØØØ		LDCØ		S
137	Ø2Ø6		1 ØØØ Ø11 ØØØ		LDC8		
138	Ø2Ø7	Ø276	Ø Ø11 111 1Ø1	CCOMP	JSB	CPRIT	GO TO PRINT
139	Ø21Ø	Ø16Ø	Ø 111 ØØØ Ø11		BRN	CCOMA	
14Ø	Ø211	Ø141	Ø 11Ø ØØ1 ØØ1	CSTM2	JSB	CCODF	
141	Ø212		1 ØØ1 Ø11 ØØØ		LDC9		OP. CODE Ø9
142	Ø213	ØØØ7	Ø ØØØ Ø11 111		BRN	CSTD3	GO TO RESTORE C- AND M- REGISTERS
143	Ø214	Ø134	Ø 1Ø1 11Ø 1Ø1	CSTE2	JSB	CCODE	GO TO LOAD OP. CODE
144	Ø215		Ø ØØ1 Ø11 ØØØ		LDC1		OP. CODE 11
145	Ø216	ØØØ7	Ø ØØØ Ø11 111		BRN	CSTD3	GO TO RESTORE C- AND M- REGISTERS
146	Ø217		Ø ØØØ 1ØØ ØØØ	LERRO	RMGRA		GO TO MAIN UNIT ERROR ROUTINE
147	Ø22Ø		Ø 1ØØ Ø1Ø ØØØ	CEXC2	ROM 2		GO TO EXCH. ROUTINE
148	Ø221		1 Ø11 Ø1Ø 1ØØ	CSTO5	YS11		WAS IT SET?
149	Ø222	ØØØ3	Ø ØØØ ØØ1 111		BRN	CSTO6	NO,GO TO ACCEPT KEYWORD
15Ø	Ø223	Ø037	Ø ØØ1 111 111		BRN	CSTO7	YES,GO TO F.BLK.SOFTWARE
151	Ø224		Ø ØØØ ØØØ ØØØ		DUMMY		
152	Ø225		Ø ØØØ ØØØ ØØØ		DUMMY		
153	Ø226		Ø Ø1Ø 1Ø1 ØØØ	CSTO1	CXM		GET ING. FLG. VECTOR
154	Ø227		Ø 1ØØ ØØ1 1ØØ		PT4		SET POINTER TO INDICATE D.S. USER CODE
155	Ø23Ø		Ø Ø11 Ø11 ØØØ		LDC3		SET FLAG TO VALUE "THREE"
156	Ø231	Ø321	1 1Ø1 ØØØ 111		BRN	CACP4	GO TO COMMON DATA ADJUSTMENT
157	Ø232		Ø Ø11 Ø1Ø Ø1Ø	CDSM9	WP,ZTC		CLEAR OP. AND USERS CODES
158	Ø233		Ø Ø1Ø ØØ1 1ØØ		PT2		PREPARE OP.CODE GEN.
159	Ø234		Ø 11Ø Ø11 ØØØ		LDC6		LOAD OP.CODE FOR STORE DIRECT
16Ø	Ø235	Ø075	Ø Ø11 11Ø 111		BRN	CDSM1	
161	Ø236		Ø Ø1Ø 1Ø1 ØØØ	CINC1	CXM		GET INT. FLG. VECTOR
162	Ø237		Ø 1ØØ ØØ1 1ØØ		PT4		SET POINTER TO INDICATE D.S. USER CODE
163	Ø24Ø		Ø 1ØØ Ø11 ØØØ		LDC4		4
164	Ø241	Ø321	1 1Ø1 ØØØ 111		BRN	CACP4	GO TO COMMON DATA ADJUSTMENT
165	Ø242		Ø Ø1Ø 1Ø1 ØØØ	CDEC1	CXM		GET INT. FLG. VECTOR
166	Ø243		Ø 1ØØ ØØ1 1ØØ		PT4		SET POINTER TO INDICATE D.S. USER CODE
167	Ø244		Ø 1Ø1 Ø11 ØØØ		LDC5		5
168	Ø245	Ø351	1 1Ø1 ØØØ 111		BRN	CACP4	GO TO COMMON DATA ADJUSTMENT
169	Ø246		Ø Ø1Ø 1Ø1 ØØØ	CACU7	CXM		GET INT. FLG. VECTOR; SAVE C-REG
17Ø	Ø247		Ø 1ØØ ØØ1 1ØØ		PT4		PREPARE D.S. USER CODE FOR TEST
171	Ø25Ø		Ø 1Ø1 1ØØ Ø1Ø		P,CM1C		IS IT ZERO?
172	Ø251	Ø161	Ø 111 ØØØ 111		BRN	CDSC1	NO;

DATA STORAGE UNIT LISTING (ROM C—0) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
173	0252	0106	0	100	011	011	CEND9	BRN	CSUPER	GO TO SUPERVISOR
174	0253		1	000	101	110	CDIGA	W,BXC		STORE DATA IN B-REG
175	0254		0	011	001	110		W,ZTC		CLEAR C-REG
176	0255		0	000	001	100		PT0		
177	0256		1	110	100	010		P,AXC		GET ENTERED DIGIT INTO C-REG
178	0257		0	110	001	110		W,CTA		COPY IT BACK TO A-REG ALSO
179	0260		1	010	101	000		MTC		PREPARE INTERNAL FLG. VECTOR FOR TEST
180	0261	0024	0	001	010	011		BRN	CDIGB	
181	0262		0	000	011	000	CDSC5	LDC0		RESET USERS CODE
182	0263		0	010	101	000		CXM		STORE M-REG; RESTORE C-REG
183	0264		1	011	001	100		PT11		LOAD SYMBOLS
184	0265		1	010	011	000		LDC10		I
185	0266		0	111	011	000		LDC7		—
186	0267	0354	1	110	110	011		BRN	CDINC	
187	0270		1	000	101	110	CBLNK	W,BXC		SAVE DATA IN B-REG TEMPORARILY
188	0271		0	011	001	110		W,ZTC		CLEAR C-REG
189	0272		0	111	110	000		CTT		CLEAR T-REG
190	0273		0	101	000	000		IS2		
191	0274		1	101	001	000		TDEC		LOAD T-REG FULL WITH ONE'S
192	0275		1	001	000	000		IS1		
193	0276		0	111	111	000		TTC		COPY ALL ONE'S TO C-REG ALSO
194	0277		1	011	001	100		PT11		PREPARE LOAD CODES FOR SYMBOLS
195	0300		0	000	110	000		RETURN		END OF THE SUBROUTINE
196	0301	0267	1	011	100	001	CREC1	JSB	CBLNK	GO TO GENERATE BLANKS
197	0302		1	011	011	000		LDC11		LOAD SYMBOL R
198	0203		0	100	011	000		LDC4		LOAD SYMBOL -
199	0304	0076	0	011	111	101		JSB	CPRIT	GO TO PRINT
200	0305		0	011	001	100		PT3		PREPARE OP. CODE MODIFICATION
201	0306		1	110	110	010		WP,AXC		OLD TO A-REG
202	0307		0	011	010	010		WP,ZTC		CLEAR C-REG
203	0310		0	001	011	000		LDC1		LOAD 1ST DIGIT
204	0311		0	010	011	000		LDC2		LOAD 2ND DIGIT
205	0312		0	011	001	100		PT3		RESTORE POINTER
206	0313		0	111	010	010		WP,APCC		INCREMENT OP. CODE BY 12
207	0314		0	010	101	000		CXM		STORE UPDATED INT. FLG. VECTOR
208	0315		0	010	001	110	CSUPE9	W,BTC		RESTORE DATA IN C-REG
209	0316		0	000	110	100		CLS		CLEAR ALL STATUS BITS
210	0317		0	011	000	100		SS3		SET OPERATION SYNTAX FLAG
211	0320	0105	0	100	010	111		BRN	CSUPE1	GO TO SUPERVISOR
212	0321		0	010	101	000	CACP4	CXM		STORE UPDATED INT. FLG. VECTOR
213	0322		0	000	100	000		RMGRA		GO TO MAIN UNIT SOFTWARE
214	0323		1	010	101	000	CPRI3	MTC		GET INT.FLG.VECTOR
215	0324		0	000	110	000		RETURN		END OF THE SUBROUTINE
216	0325		0	100	010	000	CPSUB	ROM 2		RETURN AFTER DATA ADJUSTMENT FOR PRINT
217	0326		0	100	010	000	CPLIS	ROM 2		
218	0327		0	010	010	000	CPCLR	ROM 1		GO TO CLEAR DATA ROUTINE
219	0330	0236	1	001	111	011	CINCR	BRN	CINC1	PROGRAM INCEMENT START
220	0331	0242	1	010	001	011	CDECR	BRN	CDEC1	PROGRAM DECREMENT START
221	0332		0	100	010	000	CPSTO	ROM 2		
222	0333	0040	0	010	000	011	CPSTA	BRN	CSTA1	PRGM. RUN STORE -ADD
223	0334	0050	0	010	100	011	CPSTS	BRN	CSTS1	PRGM. RUN STORE -SUB
224	0335	0013	0	000	101	111	CPSTM	BRN	CSTM1	PRGM. RUN STORE-MUL
225	0336	0012	0	000	101	011	CPSTD	BRN	CSTD1	PRGM. RUN STORE-DIV
226	0337	0301	1	100	000	111	CPRCL	BRN	CREC1	PROGRAM RUN RCL() BEGINNING
227	0340	0220	1	001	000	011	CEXC1	BRN	CEXC2	EXCH() ROUTINE BEGINNING
228	0341		0	100	000	100	CDECT	SS4		DECREMENT AND TEST BEGIN,SET TEST FLAG
229	0342		0	010	100	100		RS2		RESET STATUS ONE IF SET
230	0343	0356	1	110	111	011		BRN	CDCT1	
231	0344		0	101	100	010	CDSC4	P,CM1C		IS IT FOUR
232	0345	0262	1	011	001	011		BRN	CDSC5	NO; IT IS FIVE

DATA STORAGE UNIT LISTING (ROM C—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
233	Ø346		Ø ØØ1 ØØØ 1ØØ	SS1		SET AUX. FLAG
234	Ø347		Ø ØØØ Ø11 ØØØ	LDCØ		YES; RESET USERS CODE
235	Ø35Ø		Ø Ø1Ø 1Ø1 ØØØ	CXM		STORE UPDATED INT. FLG. V; RESTORE C-REG
236	Ø351		1 Ø11 ØØ1 1ØØ	PT11		LOAD SYMBOLS
237	Ø352		1 Ø1Ø Ø11 ØØØ	LDC1Ø		I
238	Ø353		Ø ØØØ Ø11 ØØØ	LDCØ		+
239	Ø354	Ø076	Ø Ø11 111 1Ø1	JSB	CPRIT	GO TO PRINT
240	Ø355		Ø Ø1Ø ØØ1 11Ø	W,BTC		RESTORE ADDRESS IN C-REG
241	Ø356		1 ØØ1 11Ø ØØØ	ATDS		CALL D.S.
242	Ø357		1 Ø11 Ø1Ø 1ØØ	YS11		FOUND?
243	Ø36Ø	Ø217	1 ØØØ 111 111	BRN	LERRO	NO
244	Ø361		1 Ø11 1ØØ 1ØØ	RS11		YES; RESET EXTERNAL FLAG
245	Ø362		1 Ø11 111 ØØØ	DSTC		GET DATA
246	Ø363		Ø 11Ø ØØ1 11Ø	W,CTA		COPY DATA INTO A-REG
247	Ø364		Ø ØØ1 Ø1Ø 1ØØ	YS1		CHECK AUX. FLAG
248	Ø365	Ø116	Ø 1ØØ 111 Ø11	BRN	CDEC2	NOT SET
249	Ø366	Ø146	Ø 11Ø Ø11 Ø11	BRN	CCINØ	GO TO INCREMENT SUBROUTINE
250	Ø367	Ø242	1 Ø1Ø ØØ1 Ø11	CMDEC BRN	CDEC1	GO TO COMMON ROUTINE
251	Ø37Ø		Ø Ø1Ø 1Ø1 ØØØ	CPRT1 CXM		RESTORE RIGHT HALF OF THE PRINTED MASK
252	Ø371		Ø Ø1Ø Ø1Ø ØØØ	ROM 1		GO TO PRINT SUBROUTINE
253	Ø372		1 1ØØ ØØ1 1ØØ	CRYPT PT12		PREPARE OP. MODE FLAG TEST
254	Ø373		Ø 11Ø 1ØØ Ø1Ø			IS IT ZERO?
255	Ø374	Ø37Ø	1 111 1ØØ Ø11	BRN	CPRT1	YES .
256	Ø375		Ø Ø1Ø 1Ø1 ØØØ	CXM		RESTERE INT.FLG.VECTOR IN M-REG
257	Ø376	Ø323	1 1Ø1 ØØ1 111	BRN	CPRI3	
258	Ø377	Ø236	1 ØØ1 111 Ø11	CMINC BRN	CINC1	GO TO COMMON ROUTINE

DATA STORAGE UNIT LISTING (ROM C—1)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 101 010 100	CPRNT	YS5		CHECK IF PRINT OR NOT
4	0001	0207	1 000 011 111		BRN	CPRIN	PRINT
5	0002	0227	1 001 011 111		BRN	CPRT9	DO NOT PRINT
6	0003		0 000 101 000	CLISA	DSTO		DISPLAY OLD DATA: USE OLD MASK
7	0004		0 000 100 100	CLIS2	RS0		
8	0005		0 000 010 100		YS0		IS OLD KEY STILL DOWN?
9	0006	0100	0 100 000 011		BRN	CLIS3	NO; CONTINUE
10	0007	0004	0 000 010 011		BRN	CLIS2	YES,WAIT
11	0010	0003	0 000 001 111		BRN	CLISA	FALSE KEY
12	0011		0 000 000 000		DUMMY		
13	0012	0003	0 000 001 111		BRN	CLISA	FALSE KEY
14	0013	0003	0 000 001 111		BRN	CLISA	FALSE KEY
15	0014	0003	0 000 001 111		BRN	CLISA	FALSE KEY
16	0015		0 000 000 000		DUMMY		
17	0016	0003	0 000 001 111		BRN	CLISA	FALSE KEY
18	0017		0 000 000 000		DUMMY		
19	0020	0363	1 111 001 111	CEQL1	BRN	CEQL2	GO TO CHECK AUX. FLAGS
20	0021		0 000 000 000		DUMMY		
21	0022	0003	0 000 001 111		BRN	CLISA	FALSE KEY
22	0023		0 100 000 100	CCLR1	SS4		SET CLEAR DATA ROUTINE AUXILIARY FLAG
23	0024	0003	0 000 001 111		BRN	CLISA	GO TO DISPLAY OLD DATA
24	0025		0 000 000 000		DUMMY		
25	0026	0003	0 000 001 111		BRN	CLISA	FALSE KEY
26	0027		0 011 001 110	CCLR2	W,ZTC		SET ZERO TO C-REG
27	0030		1 011 110 000		DTDS		LOAD ZERO INTO GIVEN D.S. REGISTER
28	0031	0171	0 111 100 111		BRN	CCOM60	GO TO COMMON PART WITH LIST ROUTINE
29	0032	0003	0 000 001 111		BRN	CLISA	FALSE KEY
30	0033	0003	0 000 001 111		BRN	CLISA	FALSE KEY
31	0034	0003	0 000 001 111		BRN	CLISA	FALSE KEY
32	0035		0 000 000 000		DUMMY		
33	0036	0003	0 000 001 111		BRN	CLISA	FALSE KEY
34	0037		0 000 000 000		DUMMY		
35	0040	0376	1 111 111 011	CMINC	BRN	CINC10	MANUAL INCREMENT START
36	0041	0121	0 101 000 111		BRN	CCOM1	GO TO COMMON PART OF BOTH ROUTINES
37	0042	0003	0 000 001 111		BRN	CLISA	FALSE KEY
38	0043	0003	0 000 001 111		BRN	CLISA	FALSE KEY
39	0044	0003	0 000 001 111		BRN	CLISA	FALSE KEY
40	0045		0 000 000 000		DUMMY		
41	0046	0003	0 000 001 111		BRN	CLISA	FALSE KEY
42	0047		0 000 000 000		DUMMY		
43	0050	0366	1 111 011 011	CMDEC	BRN	CDEC10	MANUAL DECREMENT START
44	0051	0243	1 010 001 111	CINTS	BRN	CEND5	GO TO ROUTINE END
45	0052	0003	0 000 001 111		BRN	CLISA	FALSE KEY
46	0053	0003	0 000 001 111		BRN	CLISA	FALSE KEY
47	0054	0003	0 000 001 111		BRN	CLISA	FALSE KEY
48	0055		0 000 000 000		DUMMY		
49	0056	0003	0 000 001 111		BRN	CLISA	FALSE KEY
50	0057		1 000 010 100	CEQL3	YS8		IS LIST DAT ROUTINE FLAG SET ?
51	0060	0003	0 000 001 111		BRN	CLISA	NO; FALSE KEY SEQUENCE
52	0061	0071	0 011 100 111		BRN	CCOM10	GO TO CHECK IF KEY IS STILL DOWN

DATA STORAGE UNIT LISTING (ROM C—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0003	0 000 001 111		BRN	CLISA	FALSE KEY
54	0063	0003	0 000 001 111		BRN	CLISA	FALSE KEY
55	0064	0003	0 000 001 111		BRN	CLISA	FALSE KEY
56	0065		0 000 000 000		DUMMY		
57	0066	0003	0 000 001 111		BRN	CLISA	FALSE KEY
58	0067		0 000 000 000		DUMMY		
59	0070	0003	0 000 001 111		BRN	CLISA	FALSE KEY
60	0071		0 100 010 000	CCOM10	ROM 2		GO TO TEST ON ROM 2
61	0072	0003	0 000 001 111		BRN	CLISA	FALSE KEY
62	0073		0 100 010 000	CRES1	ROM 2		GO TO ROM 2 FOR RESET ROUTINE START
63	0074		0 100 100 100	CEQL9	RS4		RESET CLEAR DATA FLAG
64	0075	0071	0 011 100 111		BRN	CCOM10	GO TO CHECK IF KEY IS STILL DOWN
65	0076		1 000 000 100	CLIS6	SS8		SET LIST ROUTINE AUXILIARY FLAG
66	0077	0003	0 000 001 111		BRN	CLISA	GO TO DISPLAY OLD DATA
67	0100		0 101 000 000	CLIS3	IS2		
68	0101		0 100 100 000		RBL		RESET BUSY LIGHT
69	0102		1 001 000 000		IS1		
70	0103		0 000 010 100	CLIS4	YS0		NEW KEY DOWN?
71	0104	0103	0 100 001 111		BRN	CLIS4	NO; WAIT
72	0105		1 001 001 100		PT9		
73	0106		0 000 111 100	CLIS5	PLS		WAIT FOR 3 MS.
74	0107		1 100 101 100		YP12		FINISHED?
75	0110	0106	0 100 011 011		BRN	CLIS5	NO, WAIT
76	0111		1 000 101 000		DSOF		YES; START EXECUTION
77	0112		1 010 100 000		SBL		SET BUSY LIGHT
78	0113		0 011 010 000		TKRA		ACCEPT KEYCODE
79	0114		0 000 000 000		DUMMY		
80	0115	0274	1 011 110 101	CCCIN	JSB	CINCR	GO TO INCREMENT SUBROUTINE
81	0116		0 000 010 000	CINDE	ROM 0		BACK TO COMMON FINAL PART
82	0117	0270	1 011 100 101	CCCDE	JSB	CDECR	GO TO DECREMENT SUBROUTINE
83	0120	0246	1 010 011 011		BRN	CINDT	
84	0121		0 100 101 000	CCOM1	CTS		SAVE CURRENT C-REG AND D-REG IN STACK
85	0122		1 000 010 100		YS8		IS THIS LIST ROUTINE ?
86	0123	0142	0 110 001 011		BRN	CCOM4	NO;
87	0124		1 101 001 100		PT13		YES; PREPARE MESSAGE
88	0125		1 111 011 000	CCOM2	LDC15		LOAD BLANKS
89	0126		0 101 101 100		YP5		FINISHED?
90	0127	0125	0 101 010 111		BRN	CCOM2	NO
91	0130		1 011 011 000	CCOM3	LDC11		YES; LOAD "DATA
92	0131		0 001 101 100		YP1		FINISHED?
93	0132	0130	0 101 100 011		BRN	CCOM3	NO
94	0133		1 111 011 000		LDC15		YES; LOAD BLANK
95	0134		1 111 011 000		LDC15		BLANK
96	0135		0 111 110 000		CTT		STORE LEFT HALF IN T-REG
97	0136		1 010 001 100		PT10		
98	0137		0 010 011 000		LDC2		LOAD =
99	0140	0206	1 000 011 101		JSB	CPRIN	GO TO PRINT ROUTINE
100	0141	0200	1 000 000 101		JSB	CPAPA	PAPER ADVANCE
101	0142		0 010 101 000	CCOM4	CXM		PREPARE REGISTER COUNTER; STEP ONE
102	0143		0 000 001 100		PT0		PREPARE REGISTER COUNTER; STEP TWO
103	0144		0 011 000 010		P,ZTC		RESET REGISTER COUNTER
104	0145		1 011 101 110		W,ZTA		PREPARE D.S. ADDRESS ZERO IN A-REG
105	0146		0 010 101 000	CCOM5	CXM		STORE UPDATED REGISTER COUNTER
106	0147		1 110 101 110		W,AXC		GET NEW D.S. ADDRESS INTO C-REG
107	0150		0 100 101 000		CTS		COPY NEW ADDRESS INTO STACK
108	0151		1 001 110 000		ATDS		CALL D.S.
109	0152		1 011 010 100		YS11		CHECK IF FOUND
110	0153	0230	1 001 100 011		BRN	CEND1	NO; GO TO END OF THE ROUTINE
111	0154		1 011 100 100		RS11		YES; RESET EXTERNAL FLAG
112	0155		1 000 010 100		YS8		IS THIS LIST ROUTINE ?

DATA STORAGE UNIT LISTING (ROM C—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
113	0156	0027	0	001	011	111		BRN	CCLR2	NO; GO TO CLEAR
114	0157		0	100	010	000		ROM 2		YES; GO TO KEY-DOWN CHECK
115	0160		0	010	101	000		CXM		GET INT. FLG. VECTOR
116	0161		0	100	001	100		PT4		SET POINTER TO INDICATE D.S. USER FLAG
117	0162		0	001	011	000		LDC1		SET IT TO "ONE"
118	0163		0	000	010	000		ROM 0		GO TO ROM 0 TO CALL GROUP A
119	0164		0	000	011	000	CCOMA	LDC0		RESET D.S. USER CODE TO ZERO
120	0165		0	010	101	000		CXM		STORE INT. FLG. VECTOR; RESTORE C-REG
121	0166		1	010	001	100		PT10		PREPARE SYMBOLS TO BE LOADED
122	0167		1	111	011	000		LDC15		BLANK
123	0170	0206	1	000	011	101		JSB	CPRIN	GO TO PRINT LISTED DATA
124	0171		0	110	101	000	CCOM60	STA		RESTORE OLD ADDRESS IN A-REG
125	0172	0274	1	011	110	101	CCOM6	JSB	CINCR	GO TO ADDRESS INCREMENT
126	0173		1	010	101	000		MTC		GET INT. FLG. VECTOR
127	0174		0	000	001	100		PT0		PREPARE REGISTER COUNTER
128	0175		0	111	100	010		P,CP1C		INCREMENT AND CHECK IF CARRY
129	0176	0146	0	110	011	011		BRN	CCOM5	NO CARRY; GO BACK TO LOOP
130	0177	0200	1	000	000	101		JSB	CPAPA	CARRY; GO TO PAPER ADVANCE
131	0200	0146	0	110	011	011		BRN	CCOM5	GO BACK TO LOOP
132	0201		0	101	110	000	CPAPA	PRE		PRINTER ENABLE
133	0202		1	011	010	100	CPAP1	YS11		FLAG?
134	0203	0202	1	000	001	011		BRN	CPAP1	NO;WAIT
135	0204		1	011	100	100		RS11		YES; RESET EXTERNAL FLAG
136	0205		0	011	110	000		ADV		PAPER ADVANCE
137	0206		0	000	110	000		RETURN		END OF THE SUBROUTINE
138	0207		0	100	001	100	CPRIN	PT4		PREPARE SECTOR COUNTER
139	0210		0	101	110	000		PRE		PRINTER ENABLE
140	0211		1	011	010	100	CPRT4	YS11		FLAG?
141	0212	0211	1	000	100	111		BRN	CPRT4	NO; WAIT
142	0213		1	011	100	100		RS11		YES
143	0214		1	111	110	000		CCS		RIGHT PART OF THE MASK
144	0215		1	101	110	000		TCS		LEFT PART OF THE MASK
145	0216		0	000	111	100		PLS		INCREMENT SECTOR COUNTER
146	0217		0	001	101	100		YP1		LAST SECTOR ?
147	0220	0211	1	000	100	111		BRN	CPRT4	NO
148	0221		1	011	010	100	CPRT2	YS11		YES; FLAG?
149	0222	0221	1	001	000	111		BRN	CPRT2	NO; WAIT!
150	0223		1	011	100	100		RS11		YES; RESET EXTERNAL FLAG
151	0224		0	011	110	000		ADV		PAPER ADVANCE
152	0225		1	000	010	100		YS8		IS THIS LIST ROUTINE ?
153	0226	0375	1	111	110	111		BRN	CPRT3	NO
154	0227		0	000	110	000	CPRT9	RETURN		YES; END OF THE SUBROUTINE
155	0230		1	101	001	100	CEND1	PT13		PREPARE END MESSAGE
156	0231		1	000	010	100		YS8		IS THIS LIST ROUTINE ?
157	0232	0252	1	010	101	011		BRN	CCLR3	NO
158	0233		1	111	011	000	CEND2	LDC15		YES; LOAD BLANKS
159	0234		0	010	101	100		YP2		FINISHED?
160	0235	0233	1	001	101	111		BRN	CEND2	NO
161	0236		1	100	011	000	CEND3	LDC12		YES, LOAD "END"
162	0237		1	111	101	100		YP15		FINISHED ?
163	0240	0236	1	001	111	011		BRN	CEND3	NO
164	0241		0	111	110	000		CTT		LEFT HALF TO T-REG
165	0242	0206	1	000	011	101	CEND4	JSB	CPRIN	GO TO PRINT SUBROUTINE
166	0243		1	000	100	100	CEND5	RS8		RESET AUXILIARY FLAG IS SET
167	0244		1	010	101	000		MTC		GET INT. FLG. VECTOR TO BE RESTORED
168	0245		0	100	010	000		ROM 2		GO TO AUX.FLAG RESET
169	0246		0	100	010	100	CINDT	YS4		IS TEST PROGRAMMED ?
170	0247	0116	0	100	111	011		BRN	CINDE	NO
171	0250		0	100	100	100		RS4		YES,RESET TEST FLAG
172	0251	0377	1	111	111	111		BRN	CINT1	

DATA STORAGE UNIT LISTING (ROM C—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0252		1 111 011 000	CCLR3	LDC15	LOAD BLANKS
174	0253		0 101 101 100		YP5	FINISHED?
175	0254	0252	1 010 101 011		BRN CCLR3	NO
176	0255		1 011 011 000	CCLR4	LDC11	YES: LOAD "DATA"
177	0256		0 001 101 100		YP1	FINISHED?
178	0257	0255	1 010 110 111		BRN CCLR4	NO
179	0260		0 001 011 000		LDC1	YES; LOAD "C"
180	0261		0 111 011 000		LDC7	LOAD"L"
181	0262		0 111 110 000		CTT	LEFT HALF TO T-REG
182	0263		1 101 001 100		PT13	SET POINTER
183	0264		1 011 011 000		LDC11	E
184	0265		1 000 000 100		SS8	SET AUX.FLG. TO GET OUT PRINT ROUTE
185	0266		1 100 011 000		LDC12	A
186	0267		1 011 011 000		LDC11	R
187	0270	0242	1 010 001 011		BRN CEND4	GO TO COMMON PRINT WITH LIST ROUTINE
188	0271		0 011 001 110	CDECR	W,ZTC	CLEAR C-REG
189	0272		0 011 111 110		S,ZNCC	LOAD MINUS FOR SIGN OF MANTISSA
190	0273		1 100 001 100		PT12	SET POINTER; ELIMINATE CARRY
191	0274	0277	1 011 111 111		BRN CINC1	GO TO COMMON PART WITH INCREMENT
192	0275		0 011 001 110	CINCR	W,ZTC	CLEAR C-REG
193	0276		1 100 001 100		PT12	PREPARE LOAD ONE
194	0277		0 111 100 010	CINC1	P,CP1C	GENERATE "ONE" IN SCI-NOT
195	0300		0 000 101 110		W,ZTB	
196	0301		1 111 111 010		XS,AP1A	
197	0302		1 111 111 010		XS,AP1A	
198	0303		0 111 111 010		XS,CP1C	
199	0304		0 111 111 010		XS,CP1C	
200	0305		0 001 001 010		X,AMC	
201	0306	0310	1 100 100 011		BRN CINC4	
202	0307		1 110 101 110		W,AXC	
203	0310		1 110 100 110	CINC4	M,AXC	
204	0311		0 110 100 110		M,ZMC	
205	0312	0314	1 100 110 011		BRN CINC5	
206	0313		1 110 101 110		W,AXC	
207	0314		1 000 100 110	CINC5	M,BXC	
208	0315		0 001 001 010	CINC6	X,AMC	
209	0316	0331	1 101 100 111		BRN CINC2	
210	0317		1 010 001 110		W,SRB	
211	0320		1 111 101 010		X,AP1A	
212	0321		0 000 001 110		W,ZMB	
213	0322	0331	1 101 100 111		BRN CINC2	
214	0323	0315	1 100 110 111		BRN CINC6	
215	0324		1 110 100 110	CNRM2	M,AXC	GET RESULT MANTISSA INTO C-REG
216	0325		0 110 001 110		W,CTA	COPY RESULT INTO A-REG
217	0326		0 000 110 000		RETURN	
218	0327		1 000 000 100	CPRLIS	SS8	LIST IN PROGRAM RUN,SET AUG.FLG.
219	0330	0121	0 101 000 111		BRN CCOM1	GO TO COMMON PART
220	0331		0 101 111 010	CINC2	XS,CM1C	
221	0332		0 101 111 010		XS,CM1C	
222	0333		1 011 101 010		X,ZTA	
223	0334		1 101 011 110		S,AMCA	
224	0335		1 001 111 110		S,AM1	
225	0336	0353	1 110 101 111		BRN CINC3	GO TO SUBTRACTION
226	0337		1 110 001 110		W,APBA	ADD MANTISSAS
227	0340		1 011 001 110	CINC7	W,SRA	FOR OWERFLOW CASE
228	0341		0 111 101 010		X,CP1C	INCREMENT EXPONENT ACCORDINGLY
229	0342		1 100 001 100		PT12	PREPARE NORMALIZATION CHECK
230	0343		1 001 100 010	CNRM1	P,AM1	NORMALIZATION NEEDED?
231	0344	0324	1 101 010 011		BRN CNRM2	NO
232	0345		0 100 001 110		W,SLA	YES; NORMALIZATION SHIFT

DATA STORAGE UNIT LISTING (ROM C—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 101 101 010		X,CM1C		EXPONENT DECREMENT
234	0347		1 001 101 110		W,AM1		IS IT ZERO?
235	0350	0343	1 110 001 111		BRN	CNRM1	NO
236	0351		0 011 001 110		W,ZTC		SET ZERO FOR RESULT
237	0352		0 000 110 000		RETURN		YES; FINISHED
238	0353		1 000 000 110	CINC3	M,AMB		
239	0354	0357	1 110 111 111		BRN	CINC8	
240	0355		0 011 111 110		S,ZNCC		
241	0356		1 100 101 110		W,AXB		
242	0357		1 100 001 110	CINC8	W,AMBA		SUBTRACT MANTISSAS
243	0360		1 011 111 110		S,ZTA		CANCEL SIGN
244	0361		0 000 000 000		NOP		TO ELIMINATE CARRY
245	0362	0340	1 110 000 011		BRN	CINC7	GO TO NORMALIZATION
246	0363		0 100 010 100	CEQL2	YS4		IS CLEAR DATA ROUTINE FLAG SET?
247	0364	0057	0 010 111 111		BRN	CEQL3	NO; CONTINUE IN TEST
248	0365	0074	0 011 110 011		BRN	CEQL9	YES, GO TO RESET IT
249	0366		0 000 010 000	CDEC10	ROM 0		GO TO COMMON PATH
250	0367		1 011 010 100	CCPR1	YS11		ON COMMAND ONLY ?
251	0370	0000	0 000 000 011		BRN	CPRNT	NO, GO TO PRINT
252	0371	0374	1 111 110 011		BRN	CCPR2	YES, DO NOT PRINT
253	0372		1 110 100 000	CPRTA	YPOC		CHECK PRINT ON COMMAND FLAG
254	0373	0367	1 111 011 111		BRN	CCPR1	CONTINUE
255	0374		1 011 100 100	CCPR2	RS11		RESET EXTERNAL FLAG
256	0375		0 000 010 000	CPRT3	ROM 0		GO BACK TO ROM 0
257	0376		0 000 010 000	CINC10	ROM 0		GO TO COMMON PART
258	0377		0 100 010 000	CINT1	ROM 2		GO TO TEST

DATA STORAGE UNIT LISTING (ROM C-2)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		0 001 100 100	CINT2	M,CM1		IS DECREMENTED NUMBER ZERO?
4	0001	0132	0 101 101 011		BRN	CINT3	NO,CONDITION NOT MET
5	0002	0106	0 100 011 011		BRN	CINT4	YES,CONDITION MET
6	0003	0106	0 010 101 000	CCOB2	CXM		GET INT. FLG. VECTOR FOR UPDATING
7	0004		0 010 001 100		PT2		SET OP. COD TO:
8	0005		0 111 011 000		LDC7		07
9	0006	0055	0 010 110 111		BRN	CCOB9	GO TO COMMON PART
10	0007		0 010 101 000	CCOB1	CXM		GET INT. FLG. VECTORFOR UPDATING
11	0010		0 010 001 100		PT2		SET OP. CODE TO:
12	0011		1 000 011 000		LDC8		08
13	0012	0055	0 010 110 111		BRN	CCOB9	GO TO COMMON PART
14	0013	0023	0 001 001 111	CSTD2	BRN	CSTD3	STORE-DIVIDE
15	0014	0020	0 001 000 011	CSTM2	BRN	CSTM3	STORE-MULTIPLY
16	0015		0 000 000 000		DUMMY		
17	0016		0 000 000 000		DUMMY		
18	0017		0 000 000 000		DUMMY		
19	0020	0033	0 001 110 001	CSTM3	JSB	CCOB4	GO TO SET OP. CODE AND PREPARE BLANKS
20	0021		0 001 011 000		LDC1		LOAD CODE FOR *
21	0022	0377	1 111 111 111		BRN	CPRINT	GO TO PRINT
22	0023	0041	0 010 001 001	CSTD3	JSB	CCOB3	GO TO SET OP. CODE AND PREPARE BLANKS
23	0024		0 110 011 000		LDC6		LOAD CODE FOR DIVIDE
24	0025	0377	1 111 111 111		BRN	CPRNT	GO TO PRINT
25	0026	0002	0 000 001 101	CSTA3	JSB	CCOB2	GO TO SET OP. CODE AND PREPARE BLANKS
26	0027		0 000 011 000		LDC0		LOAD CODE FOR +
27	0030	0377	1 111 111 111		BRN	CPRNT	GO TO PRINT
28	0031	0006	0 000 011 101	CSTS3	JSB	CCOB1	GO TO SET OP. CODE AND PREPARE BLANKS
29	0032		0 111 011 000		LDC7		LOAD CODE FOR—
30	0033	0377	1 111 111 111		BRN	CPRNT	GO TO PRINT
31	0034		0 010 101 000	CCOB4	CXM		GET INT. FLG. VECTOR FOR UPDATING
32	0035		0 010 001 100		PT2		SET OP. CODE TO:
33	0036		1 001 011 000		LDC9		09
34	0037	0055	0 010 110 111		BRN	CCOB9	GO TO COMMON PART
35	0040		0 010 010 000	CCOMR	ROM 1		GO TO EXECUTE ROUTINE ON ROM U
36	0041	0026	0 001 011 011	CSTA2	BRN	CSTA3	STORE-ADD
37	0042		0 010 101 000	CCOB3	CXM		GET INT. FLG. VECTOR FOR UPDATING
38	0043		0 011 001 100		PT3		SET OP. CODE TO:
39	0044		0 001 011 000		LDC1		10
40	0045	0054	0 010 110 011		BRN	CCOB8	GO TO FINISH OP.CODE LOADING
41	0046		0 000 010 100	CINT1	YS0		IS A KEY DOWN?
42	0047	0156	0 110 111 011		BRN	CCOMB	NO: CONTINUE IN ROUTINE EXECUTION
43	0050		0 010 010 000		ROM 1		YES: GO TO ROUTINE END
44	0051	0031	0 001 100 111	CSTS2	BRN	CSTS3	STORE-SUBRACT
45	0052		0 000 000 000		DUMMY		
46	0053		0 000 000 000		DUMMY		
47	0054		0 000 011 000	CCOB8	LDC0		10
48	0055		0 010 101 110	CCOB9	CXM		RESTORE INT.FLG. VECTOR
49	0056		1 000 101 110		W,BXC		DATA TEMPORARILY INTO B-REG
50	0057		0 011 001 110		W,ZTC		CLEAR C-REG
51	0060		0 111 110 000		CTT		CLEAR T-REG
52	0061		0 101 000 000		IS2		

DATA STORAGE UNIT LISTING (ROM C—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062		1 101 001 000		TDEC		LOAD T-REG FULL WITH ONES
54	0063		1 001 000 000		IS1		
55	0064		0 111 111 000		TTC		COPY "ONES" INTO C-REG ALSO
56	0065		1 010 001 100		PT10		PREPARE OP.CODE LOAD
57	0066		0 101 100 100		RS5		RESET FLAT 5 IF SET
58	0067		1 000 100 100		RS8		RESET FLAG 8 IF SET
59	0070		0 000 110 000		RETURN		END OF THE SUBROUTINE
60	0071		0 000 000 000		DUMMY		
61	0072	0100	0 100 000 011		BRN	CCOMS	
62	0073		0 000 010 000	CRES3	ROM 0		GO TO MAIN UNIT
63	0074		0 000 110 100	CRES2	CLS		CLEAR ALL STATUS BITS
64	0075		0 100 000 100		SS4		SET RESET ROUTINE AUX. FLAG
65	0076		1 010 101 000		MTC		GET INT. FLG. VECTOR FOR UPDATE
66	0077	0073	0 011 101 111		BRN	CRES3	
67	0100		0 000 100 100	CCOMS	RS0		RESET KEY-DOWN FLAG
68	0101		0 000 010 100		YS0		IS THE KEY STILL DOWN
69	0102	0040	0 010 000 011		BRN	CCOMR	NO: GO AND EXECUTE
70	0103	0100	0 100 000 011		BRN	CCOMS	YES: WAIT IN THIS LOOP
71	0104		0 000 000 000		DUMMY		
72	0105		0 000 000 000		DUMMY		
73	0106		0 000 010 000	CINT4	ROM 0		GO TO CFINI
74	0107		0 000 000 000		DUMMY		
75	0110		0 000 000 000		DUMMY		
76	0111		0 000 000 000		DUMMY		
77	0112		0 000 000 000		DUMMY		
78	0113		0 000 011 000	CPLI2	LDC0		RESET IT TO ALLOW DATA PRINT
79	0114		0 001 001 100		PT1		PREPARE FLAG FOR RUN MODE INDICATION
80	0115		0 010 011 000		LDC2		SET IT
81	0116		0 010 101 000	CPLI3	CXM		STORE UPDATED VECTOR, RESTORE DATA
82	0117		1 000 000 100		SS8		SET LIST ROUTINE AUX.FLAG
83	0120		0 010 010 000		ROM 1		
84	0121		0 001 001 100	CEND7	PT1		PREPARE RUN MODE INDICATOR TEST
85	0122		0 110 100 010		P,ZMC		IS IT ZERO ?
86	0123	0127	0 101 011 111		BRN	CEND8	YES
87	0124		1 100 001 100		PT12		NO,PREPARE OP.MODE CODE LOAD
88	0125		0 010 011 000		LDC2		SET RUN MODE CODE
89	0126		0 001 001 100		PT1		PREPARE AUX.FLAGS RESET
90	0127		0 011 010 010	CEND8	WP,ZTC		AUX.FLG.RESET
91	0130		0 010 101 000		CXM		STORE UPDATED INT.FLG. VECTOR
92	0131	0247	1 010 011 111		BRN	CEND9	
93	0132		0 010 101 000	CINT3	CXM		GET INT.FLT.VECTOR
94	0133		0 110 001 100		PT6		PREPARE CONDITION FLAG TO BE SET
95	0134		0 001 011 000		LDC1		SET CONDITION NOT MET FLAG
96	0135		0 010 101 000		CXM		RESTORE DATA AND INT.FLG.VECTOR
97	0136	0106	0 100 011 011		BRN	CINT4	
98	0137		0 000 000 000		DUMMY		
99	0140		0 000 000 000		DUMMY		
100	0141		0 000 000 000		DUMMY		
101	0142		0 000 000 000		DUMMY		
102	0143		0 000 000 000		DUMMY		
103	0144		0 000 000 000		DUMMY		
104	0145		0 000 000 000		DUMMY		
105	0146		0 000 000 000		DUMMY		
106	0147		0 000 000 000		DUMMY		
107	0150		0 000 000 000		DUMMY		
108	0151		0 000 000 000		DUMMY		
109	0152		0 000 000 000		DUMMY		
110	0153		0 000 000 000		DUMMY		
111	0154		0 000 000 000		DUMMY		
112	0155		0 000 000 000		DUMMY		

DATA STORAGE UNIT LISTING (ROM C—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	0156		1 011 111 000	CCOMB	DSTC	GET THE DATA OUT
114	0157		0 010 010 000		ROM 1	GO BACK TO LIST ROUTINE ON ROM 1
115	0160		1 011 100 100	CCOMZ	RSI1	RESET EXTERNAL FLAG
116	0161	0046	0 010 011 011		BRN CINT1	GO TO CHECK FOR KEY-DOWN
117	0162		0 000 000 000		DUMMY	
118	0163		0 000 000 000		DUMMY	
119	0164		0 000 000 000		DUMMY	
120	0165		0 000 000 000		DUMMY	
121	0166		0 000 000 000		DUMMY	
122	0167		0 000 000 000		DUMMY	
123	0170		0 000 000 000		DUMMY	
124	0171		0 000 000 000		DUMMY	
125	0172		0 000 000 000		DUMMY	
126	0173		0 000 000 000		DUMMY	
127	0174		0 000 000 000		DUMMY	
128	0175		0 000 000 000		DUMMY	
129	0176		0 000 000 000		DUMMY	
130	0177		0 000 000 000		DUMMY	
131	0200		0 000 000 000		DUMMY	
132	0201		0 000 000 000		DUMMY	
133	0202		0 000 000 000		DUMMY	
134	0203		0 000 000 000		DUMMY	
135	0204		0 000 000 000		DUMMY	
136	0205		0 000 000 000		DUMMY	
137	0206		0 000 000 000		DUMMY	
138	0207		0 000 010 000	CPST3	ROM 0	GO TO EXECUTE OLD OPERATION
139	0210		0 000 000 000		DUMMY	
140	0211		0 000 000 000		DUMMY	
141	0212		0 000 000 000		DUMMY	
142	0213		0 000 000 000		DUMMY	
143	0214		0 000 000 000		DUMMY	
144	0215		0 000 000 000		DUMMY	
145	0216		0 000 000 000		DUMMY	
146	0217		0 000 000 000		DUMMY	
147	0220		0 000 000 000		DUMMY	
148	0221		0 010 101 000	CEXC3	CXM	GET INT.FL.VECTOR
149	0222		0 100 001 100		PT4	SET POINTER TO INDICATE D.S.USER CODE
150	0223		0 010 011 000		LDC2	SET IT TO TWO
151	0224	0320	1 101 000 011		BRN LEXC4	
152	0225		0 000 000 000		DUMMY	
153	0226		0 000 010 000	CPST2	ROM 0	GO TO SET USERS CODE
154	0227		0 000 000 000		DUMMY	
155	0230		0 000 000 000		DUMMY	
156	0231		0 000 000 000		DUMMY	
157	0232		0 000 000 000		DUMMY	
158	0233		0 000 000 000		DUMMY	
159	0234		0 000 000 000		DUMMY	
160	0235		0 000 000 000		DUMMY	
161	0236		0 000 000 000		DUMMY	
162	0237		0 000 000 000		DUMMY	
163	0240		0 000 000 000		DUMMY	
164	0241		0 000 000 000		DUMMY	
165	0242		0 000 000 000		DUMMY	
166	0243		0 000 000 000		DUMMY	
167	0244		0 101 100 100	CACU9	RS5	RESET "DO NOT PRINT" FLAG IF SET
168	0245		0 000 010 000		ROM 0	GO TO USERS CODE TEST
169	0246	0121	0 101 000 111	CEND6	BRN CEND7	
170	0247		1 100 101 000	CEND9	DNR	LAST ADDRESS INTO C-REG
171	0250		1 100 101 000		DNR	RESTORE ORIGINAL C- AND D-REG CONTENTS
172	0251		0 000 010 000		ROM 0	GO TO MAIN UNIT SUPERVISOR

DATA STORAGE UNIT LISTING (ROM C—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 000 000 000		DUMMY		
174	0253		0 000 000 000		DUMMY		
175	0254		0 000 000 000		DUMMY		
176	0255		0 000 000 000		DUMMY		
177	0256		0 000 000 000		DUMMY		
178	0257		0 000 000 000		DUMMY		
179	0260		0 000 000 000		DUMMY		
180	0261		0 000 000 000		DUMMY		
181	0262		0 000 000 000		DUMMY		
182	0263		0 000 000 000		DUMMY		
183	0264		0 000 000 000		DUMMY		
184	0265		0 000 000 000		DUMMY		
185	0266		0 000 000 000		DUMMY		
186	0267		0 000 000 000		DUMMY		
187	0270		0 000 000 000		DUMMY		
188	0271		0 000 000 000		DUMMY		
189	0272		0 000 000 000		DUMMY		
190	0273		0 000 000 000		DUMMY		
191	0274		0 000 000 000		DUMMY		
192	0275		0 000 000 000		DUMMY		
193	0276		0 000 000 000		DUMMY		
194	0277		0 000 000 000		DUMMY		
195	0300		0 000 000 000		DUMMY		
196	0301		0 000 000 000		DUMMY		
197	0302		0 000 000 000		DUMMY		
198	0303		0 000 000 000		DUMMY		
199	0304		0 000 000 000		DUMMY		
200	0305		0 000 000 000		DUMMY		
201	0306		0 000 000 000		DUMMY		
202	0307		0 000 000 000		DUMMY		
203	0310		0 000 000 000		DUMMY		
204	0311		0 000 000 000		DUMMY		
205	0312		0 000 000 000		DUMMY		
206	0313		0 000 000 000		DUMMY		
207	0314		0 000 000 000		DUMMY		
208	0315		0 000 000 000		DUMMY		
209	0316		0 000 000 000		DUMMY		
210	0317		0 000 000 000		DUMMY		
211	0320		0 000 010 000	LEXC4	ROM 0		GO TO CACP4
212	0321		0 000 000 000		DUMMY		
213	0322		0 000 000 000		DUMMY		
214	0323		0 000 000 000		DUMMY		
215	0324		0 000 000 000		DUMMY		
216	0325		0 000 000 000		DUMMY		
217	0326	0244	1 010 010 011	CACU8	BRN	CACU9	
218	0327		0 010 101 000	CPLI1	CXM		GET INT.FLG.VECTOR
219	0330		0 110 100 010		P,ZMC		IS OP.MODE CODE ZERO ?
220	0331	0116	0 100 111 011		BRN	CPLI3	YES
221	0332	0113	0 100 101 111		BRN	CPLI2	NO
222	0333		0 010 101 000	CPST1	CXM		GET INT.FLG.VECTOR
223	0334		1 100 001 100		PTI2		PREPARE OP.MODE TEST
224	0335		0 110 100 010		P,ZMC		IS IT ZERO
225	0336	0226	1 001 011 011		BRN	CPST2	YES,IT IS STEP MODE
226	0337		0 000 001 100		PT0		NO,PROGRAM RUN MODE
227	0340		0 110 011 000		LDC6		LOAD OP.CODE FOR STORE DIRECT
228	0341		0 010 101 000		CXM		STORE UPDATED VECTOR,RESTORE DATA
229	0342		1 000 101 110		W,BXC		SAVE DATA IN B-REG TEMPORARILY
230	0343		1 010 101 000		MTC		PREPARE INT.FLG.VECTOR FOR TEST
231	0344	0207	1 000 011 111		BRN	CPST3	
232	0345		0 000 000 000		DUMMY		

DATA STORAGE UNIT LISTING (ROM C—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
233	0346		0 000 000 000		DUMMY	
234	0347		0 000 000 000		DUMMY	
235	0350		0 000 000 000		DUMMY	
236	0351		0 000 000 000		DUMMY	
237	0352		0 000 000 000		DUMMY	
238	0353		0 000 000 000		DUMMY	
239	0354		0 000 000 000		DUMMY	
240	0355		0 000 000 000		DUMMY	
241	0356		0 000 000 000		DUMMY	
242	0357		0 000 000 000		DUMMY	
243	0360		0 000 000 000		DUMMY	
244	0361		0 000 000 000		DUMMY	
245	0362		0 000 000 000		DUMMY	
246	0363		0 000 000 000		DUMMY	
247	0364		0 000 000 000		DUMMY	
248	0365		0 000 000 000		DUMMY	
249	0366		0 000 000 000		DUMMY	
250	0367		0 000 000 000		DUMMY	
251	0370		0 000 000 000		DUMMY	
252	0371		0 000 000 000		DUMMY	
253	0372		0 000 000 000		DUMMY	
254	0373		0 000 000 000		DUMMY	
255	0374		0 000 000 000		DUMMY	
256	0375		0 000 000 000		DUMMY	
257	0376		0 000 000 000		DUMMY	
258	0377		0 000 010 000	CPRNT	ROM 0	GO TO PRINT

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—Ø)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		0	010	101	000		CXM	
4	0001		1	101	001	100		PT13	
5	0002	0005	0	000	010	111		BRN	PROGM2
6	0003		0	000	100	000	BNOCH	RMGRA	
7	0004	0003	0	000	001	111		BRN	BNOCH
8	0005		1	000	010	000	PROGM2	ROM 4	CONTINUE PROGRAM MODE IN ROM 4
9	0006		0	111	000	100	SUPCØ	SS7	SET FIRST ENTRY FLAG
10	0007	0106	0	100	011	011		BRN	SUPC
11	0010	0241	1	010	000	111	KY1	BRN	CD1 KEY 1 DOWN
12	0011	0306	1	100	011	011		BRN	DATAØ
13	0012	0242	1	010	001	011	KY2	BRN	CD2 KEY 2 DOWN
14	0013	0243	1	010	001	111	KY3	BRN	CD3 KEY 3 DOWN
15	0014	0244	1	010	010	011	KY4	BRN	CD4 KEY 4 DOWN
16	0015		0	000	000	000		DUMMY	
17	0016	0245	1	010	010	111	KY5	BRN	CD5 KEY 5 DOWN
18	0017		0	000	100	000	EQUAL	RMGRA	GO TO = ROUTINE IN ROM A
19	0020	0246	1	010	011	011	KY6	BRN	CD6 KEY 6 DOWN
20	0021		0	000	000	000		DUMMY	
21	0022	0247	1	010	011	111	KY7	BRN	CD7 KEY 7 DOWN
22	0023	0250	1	010	100	011	KY8	BRN	CD8 KEY 8 DOWN
23	0024	0251	1	010	100	111	KY9	BRN	CD9 KEY 9 DOWN
24	0025	0071	0	011	100	111		BRN	PRINTØ PRINT DATA, SAVE USERCODE
25	0026	0252	1	010	101	011	KY10	BRN	CD10 KEY 10 DOW5
26	0027		0	000	000	000		DUMMY	
27	0030	0253	1	010	101	111	KY11	BRN	CD11 KEY 11 DOWN
28	0031		0	000	000	000		DUMMY	
29	0032	0254	1	010	110	011	KY12	BRN	CD12 KEY 12 DOWN
30	0033	0255	1	010	110	111	KY13	BRN	CD13 KEY 13 DOWN
31	0034	0256	1	010	111	011	KY14	BRN	CD14 KEY 14 DOWN
32	0035		0	000	000	000		DUMMY	
33	0036	0257	1	010	111	111	KY15	BRN	CD15 KEY 15 DOWN
34	0037		0	000	000	000		DUMMY	
35	0040	0205	1	000	010	111		BRN	START FUNCTION BLOCK START
36	0041		0	100	010	100	TABØ3	YS4	IS DO NOT PRINT FLAG SET?
37	0042	0355	1	110	110	111		BRN	PRINTT NO, GO TO CHECK PRINT ON COMMAND FLAG
38	0043	0225	1	001	010	111		BRN	SUPCØ2 YES, GO TO SUPERVISOR
39	0044		0	010	101	000	TABØ2	CXM	RESTORE M VECTOR
40	0045	0225	1	001	010	111		BRN	SUPCØ2 GO TO SUPERVISOR
41	0046		0	000	000	000		DUMMY	
42	0047		1	000	010	000	EEX	ROM 4	GO TO ENTER EXP ROUTINE
43	0050		1	101	100	010		P,AM1A	IS USERCODE 1
44	0051	0053	0	010	101	111		BRN	TEMP NO, CONTINUE CHECK
45	0052	0317	1	100	111	111		BRN	DATA DATA ENTRY CHECK
46	0053		1	101	100	010	TEMP	P,AM1A	IS USERCODE 2?
47	0054	0056	0	010	111	011		BRN	TEMP1 NO, CONTINUE TEST
48	0055	0231	1	001	100	111		BRN	PRINT YES, RETURN FROM PRINTING
49	0056		1	101	100	010	TEMP1	P,AM1A	IS USERCODE 3?
50	0057	0061	0	011	000	111		BRN	TEMP2
51	0060	0140	0	110	000	011		BRN	AENTRY YES, ANGLE ENTRY CANCEL
52	0061		1	101	100	010	TEMP2	P,AM1A	IS USERCODE 4

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0064	0 011 010 011		BRN	TEMP5	NO, CONTINUE
54	0063	0047	0 010 011 111		BRN	EEX	YES, ENTER EXP ROUTINE
55	0064		1 101 100 010	TEMP5	P,AM1A		IS USERCODE 5
56	0065	0323	1 101 001 111		BRN	LSTP	NO, CHECK FOR RUN MODE
57	0066		1 010 101 000		MTC		
58	0067	0133	0 101 101 111		BRN	TEMP4	
59	0070	0103	0 100 001 111	TAB	BRN	TAB00	START OF TABLE ROUTINE
60	0071	0165	0 111 011 001	PRINT0	JSB	SAVE	SAVE OLD USERCODE
61	0072		0 111 001 100		PT7		
62	0073		0 010 011 000		LDC2		LOAD NEW USERCODE 2
63	0074		0 010 101 000		CXM		RESTORE M VECTOR
64	0075		0 000 100 000	PRINT1	RMGRA		GO TO PRINT DATA ROUTINE
65	0076		0 001 011 000	TEMP3	LDC1		YES, SET TO PROGRAM MODE
66	0077		0 111 001 100		PT7		
67	0100		0 111 011 000		LDC7		SET USERCODE TO 7
68	0101	0177	0 111 111 111		BRN	RSTR	
69	0102	0071	0 011 100 111		BRN	PRINT0	DATA PRINT FROM ROM 7
70	0103		1 011 001 100	TAB00	PT11		
71	0104	0107	0 100 011 111		BRN	TAB0	
72	0105		0 000 100 000	CAL	RMGRA		
73	0106		0 000 100 000	SUPC	RMGRA		GO TO ROM A SUPERVISOR
74	0107		0 010 101 000	TAB0	CXM		
75	0110		0 100 011 000		LDC4		LOAD 4 AS DIC
76	0111		0 001 011 000		LDC1		STORE COUNTER FOR TWO DIGIT ENTRY
77	0112		0 111 001 100		PT7		CHECK USERCODE FOR 1
78	0113		0 001 100 010		P,CM1		
79	0114	0044	0 010 010 011		BRN	TAB02	NO, CONTINUE
80	0115		0 010 101 000		CXM		
81	0116	0041	0 010 000 111		BRN	TAB03	
82	0117		1 110 010 000	START1	ROM 7		
83	0120		1 000 101 110	RESTOR	W,BXC		STORE C
84	0121		1 110 101 110		W,AXC		PUT M VECTOR IN C
85	0122		1 000 001 100		PT8		GET OLD USERCODE
86	0123		1 110 100 010		P,AXC		PUT IT IN A
87	0124		1 011 001 110		W,SRA		ALIGNMENT SHIFT
88	0125		0 000 011 000		LDC0		SET M8 TO ZERO
89	0126		0 111 001 100		PT7		
90	0127		1 110 100 010		P,AXC		RESTORE OLD USERCODE
91	0130		0 010 101 000		CXM		RESTORE M VECTOR
92	0131		0 010 001 110		W,BTC		RESTORE DATA
93	0132		0 000 110 000		RETURN		
94	0133		0 110 011 000	TEMP4	LDC6		LOAD USERCODE TO 6
95	0134		1 100 001 100		PT12		
96	0135		1 001 100 010		P,AM1		MANUAL MODE?
97	0136	0177	0 111 111 111		BRN	RSTR	NO, GO TO RESTORE
98	0137	0076	0 011 111 011		BRN	TEMP3	
99	0140		0 010 101 000	AENTRY	CXM		
100	0141		1 010 001 100		PT10		
101	0142		1 001 011 000		LDC9		
102	0143		0 010 101 000		CXM		
103	0144		1 011 101 110		W,ZTA		
104	0145	0161	0 111 001 001		JSB	DISPLY	
105	0146		0 000 100 000	SUPCB	RMGRA		
106	0147	0071	0 011 100 111		BRN	PRINT0	GO TO PRINT FROM ROM 2
107	0150		0 000 100 000	SUPCK	RMGRA		GO TO MANUAL SUPERVISOR WITHOUT DISPLAY
108	0151		0 010 101 000	PRINT2	CXM		
109	0152		0 001 001 100		PT1		LOAD PRINT RETURN FLAG
110	0153		0 000 011 000		LDC0		
111	0154		0 010 101 000		CXM		
112	0155	0071	0 011 100 111		BRN	PRINT0	

(104)

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
113	Ø156	Ø165	Ø	111	Ø11	ØØ1	ENTRY1	JSB	SAVE	SAVE OLD USERCODE
114	Ø157		1	Ø11	ØØ1	1ØØ		PT11		
115	Ø16Ø		Ø	Ø1Ø	Ø1Ø	ØØØ	ENTRY	ROM 1		ANGLE ENTRY HEAD
116	Ø161	Ø156	Ø	11Ø	111	Ø11		BRN	ENTRY1	TO SAVE USERCODE FROM ANGLE ENTRY
117	Ø162		1	11Ø	Ø1Ø	ØØØ	DISPLY	ROM 7		GO TO ANGLE DISPLAY VIA ROM 7
118	Ø163		1	11Ø	Ø1Ø	ØØØ	PRANG	ROM 7		GO TO PRINT ANGLE
119	Ø164	Ø117	Ø	1Ø1	ØØØ	ØØ1		JSB	RESTOR	ANGLE ENTRY DONE FROM ROM 2
12Ø	Ø165	Ø163	Ø	111	ØØ1	111		BRN	PRANG	GO TO PRINT IN DMS FORMAT
121	Ø166		Ø	Ø1Ø	1Ø1	ØØØ	SAVE	CXM		SAVE OLD USERCODE
122	Ø167		Ø	111	ØØ1	1ØØ		PT7		GET OLD USERCODE
123	Ø17Ø		1	11Ø	1ØØ	Ø1Ø		P,AXC		PUT IT IN A
124	Ø171		Ø	1ØØ	ØØ1	11Ø		W,SLA		ALIGNMENT SHIFT
125	Ø172	Ø174	Ø	111	11Ø	Ø11		BRN	SAVE1	
126	Ø173		1	1ØØ	Ø1Ø	ØØØ	A2	ROM 6		RETURN AFTER DATA PRINT TO ROM 6
127	Ø174		1	ØØØ	ØØ1	1ØØ	SAVE1	PT8		SAVE OLD USERCODE
128	Ø175		1	11Ø	1ØØ	Ø1Ø		P,AXC		
129	Ø176		Ø	ØØØ	11Ø	ØØØ		RETURN		
13Ø	Ø177		Ø	Ø1Ø	1Ø1	ØØØ	RSTR	CXM		
131	Ø2ØØ		1	1ØØ	1Ø1	ØØØ		DNR		RESTORE C REGISTER
132	Ø2Ø1		Ø	ØØØ	Ø1Ø	1ØØ	WAIT	YSØ		
133	Ø2Ø2	Ø227	1	ØØ1	Ø11	111		BRN	CALØ	
134	Ø2Ø3		Ø	ØØØ	1ØØ	1ØØ		RSØ		
135	Ø2Ø4	Ø2Ø1	1	ØØØ	ØØØ	111		BRN	WAIT	WAIT FOR KEY UP
136	Ø2Ø5		1	Ø11	1ØØ	1ØØ	START	RS11		
137	Ø2Ø6		1	Ø1Ø	Ø1Ø	1ØØ		YS1Ø		IS SHIFT KEY DOWN
138	Ø2Ø7	Ø215	1	ØØØ	11Ø	111		BRN	FAD	NO, CONTINUE
139	Ø21Ø		Ø	1Ø1	ØØØ	ØØØ		IS2		YES, CALL CHANNEL 3
14Ø	Ø211	Ø177	1	ØØØ	ØØØ	ØØ1		LIO	4ØØ	
141	Ø212		Ø	11Ø	1ØØ	ØØØ		GIOE		
142	Ø213		1	ØØ1	ØØØ	ØØØ		IS1		CHANNEL 3 NOT THERE
143	Ø214	Ø117	Ø	1ØØ	111	111		BRN	START1	
144	Ø215		1	Ø11	1Ø1	11Ø	FAD	W,ZTA		CLEAR A PREPARE FOR LABEL
145	Ø216		1	Ø1Ø	1ØØ	ØØØ		SBL		SET BUSY LIGHT
146	Ø217		Ø	Ø11	Ø1Ø	ØØØ		TKRA		
147	Ø22Ø		1	Ø1Ø	1Ø1	ØØØ	CAL1	MTC		
148	Ø221		1	11Ø	1Ø1	11Ø		W,AXC		
149	Ø222	Ø1Ø5	Ø	1ØØ	Ø1Ø	111		BRN	CAL	
15Ø	Ø223		Ø	ØØØ	1ØØ	ØØØ	BET	RMGRA		RETURN TO ROM A
151	Ø224	Ø117	Ø	1Ø1	ØØØ	ØØ1	SUPCØ1	JSB	RESTOR	RESTORE OLD ISERCODE
152	Ø225		Ø	ØØØ	11Ø	1ØØ	SUPCØ2	CLS		CLEAR STATUS
153	Ø226	ØØØ6	Ø	ØØØ	Ø11	Ø11		BRN	SUPCØ	GO TO CLEAR STATUS BITS
154	Ø227		Ø	11Ø	ØØ1	11Ø	CALØ	W,CTA		
155	Ø23Ø	Ø22Ø	1	ØØ1	ØØØ	Ø11		BRN	CAL1	
156	Ø231		Ø	ØØ1	ØØ1	1ØØ	PRINT	PT1		CHECK WHERE TO RETURN TO AFTER PRINT
157	Ø232		1	1Ø1	1ØØ	Ø1Ø		P,AM1A		
158	Ø233	Ø235	1	ØØ1	11Ø	111		BRN	AØ	
159	Ø234	Ø224	1	ØØ1	Ø1Ø	Ø11		BRN	SUPCØ1	
16Ø	Ø235		1	1Ø1	1ØØ	Ø1Ø	AØ	P,AM1A		
161	Ø236	Ø224	1	ØØ1	Ø1Ø	Ø11		BRN	SUPCØ1	
162	Ø237	Ø342	1	11Ø	ØØ1	Ø11		BRN	A1	
163	Ø24Ø	Ø375	1	111	11Ø	111		BRN	PROGM	PROGRAM MODE TABLE ROUTINE
164	Ø241		1	111	1ØØ	11Ø	CD1	M,AP1A		SET A REG ACCORDING
165	Ø242		1	111	1ØØ	11Ø	CD2	M,AP1A		TO FUNCTION DESIRED
166	Ø243		1	111	1ØØ	11Ø	CD3	M,AP1A		
167	Ø244		1	111	1ØØ	11Ø	CD4	M,AP1A		
168	Ø245		1	111	1ØØ	11Ø	CD5	M,AP1A		
169	Ø246		1	111	1ØØ	11Ø	CD6	M,AP1A		
17Ø	Ø247		1	111	1ØØ	11Ø	CD7	M,AP1A		
171	Ø25Ø		1	111	1ØØ	11Ø	CD8	M,AP1A		
172	Ø251		1	111	1ØØ	11Ø	CD9	M,AP1A		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	Ø252		1 111 100 110	CD1Ø	M,AP1A	
174	Ø253		1 111 100 110	CD11	M,AP1A	
175	Ø254		1 111 100 110	CD12	M,AP1A	
176	Ø255		1 111 100 110	CD13	M,AP1A	
177	Ø256		1 111 100 110	CD14	M,AP1A	
178	Ø257		1 111 100 110	CD15	M,AP1A	
179	Ø26Ø		Ø 100 101 000		CTS	SAVE C AND D REG IN STACK
18Ø	Ø261		1 010 101 000		MTC	GET INT. FLAG VECTOR
181	Ø262		Ø 111 001 100		PT7	SET POINTER
182	Ø263		Ø 101 011 000		LDC5	SET USERCODE TO 5
183	Ø264		Ø 010 101 000		CXM	RESTORE INT. FLAG VECTOR
184	Ø265		Ø 011 001 110		W,ZTC	CLEAR C REG
185	Ø266		Ø 110 001 100		PT6	SET POINTER
186	Ø267		Ø 001 011 000		LDC1	SET MANTISSA TO 1ØØØ
187	Ø27Ø		1 110 101 110		W,AXC	
188	Ø271		1 101 000 110		M,AMCA	CALCULATE DESIRED LABEL
189	Ø272		Ø 100 001 110	ALP	W,SLA	
19Ø	Ø273		Ø ØØØ 111 100		PLS	
191	Ø274		1 100 101 100		YP12	IS POINTER = 12?
192	Ø275	Ø272	1 Ø11 101 Ø11		BRN	ALP NO, CONTINUE
193	Ø276		1 110 101 110		W,AXC	RETURN LABEL TO C REG
194	Ø277		Ø 111 101 110		W,CP1C	ADD EXPONENET
195	Ø3ØØ		Ø 111 101 110		W,CP1C	
196	Ø3Ø1		Ø 101 ØØØ ØØØ		IS2	
197	Ø3Ø2		1 ØØ1 ØØ1 ØØØ		PDEC	
198	Ø3Ø3		1 ØØ1 ØØ1 ØØØ		PDEC	
199	Ø3Ø4		1 ØØ1 ØØØ ØØØ		ISI	
2ØØ	Ø3Ø5	Ø223	1 ØØ1 ØØ1 111		BRN	BET PERFORM A JSBLC
2Ø1	Ø3Ø6	Ø165	Ø 111 Ø11 ØØ1	DATAØ	JSB	SAVE SAVE OLD USERCODE
2Ø2	Ø3Ø7		Ø 111 ØØ1 100		PT7	
2Ø3	Ø31Ø		Ø ØØ1 Ø11 ØØØ		LDC1	SET USERCODE TO 1
2Ø4	Ø311		1 100 ØØ1 100		PT12	
2Ø5	Ø312		Ø ØØØ Ø11 ØØØ		LDCØ	SET TO MANUAL MODE
2Ø6	Ø313		Ø Ø1Ø 101 ØØØ		CXM	
2Ø7	Ø314		Ø Ø11 ØØ1 110		W,ZTC	CLEAR C FOR DATA ENTRY
2Ø8	Ø315		Ø 111 ØØØ 100		SS7	SET FIRST ENTRY FLAG
2Ø9	Ø316	Ø146	Ø 110 Ø11 Ø11		BRN	SUPCB GO TO MANUAL SUPERVISOR
21Ø	Ø317		1 100 ØØ1 100	DATA	PT12	
211	Ø32Ø		1 ØØ1 100 Ø10		P,AM1	CHECK FOR MANUAL MODE
212	Ø321	Ø224	1 ØØ1 Ø10 Ø11		BRN	SUPCØ1 NO, END DATA ENTRY
213	Ø322	Ø146	Ø 110 Ø11 Ø11		BRN	SUPCB CONTINUE DATA ENTRY
214	Ø323		1 100 ØØ1 100	LSTP	PT12	
215	Ø324		1 ØØ1 100 Ø10		P,AM1	ARE WE IN MANUAL MODE
216	Ø325	Ø1Ø5	Ø 100 Ø10 111		BRN	CAL GO TO SUPERVIXOR SKIP USERCODE CHECK
217	Ø326		Ø 111 ØØ1 100		PT7	YES
218	Ø327		Ø 100 101 ØØØ		CTS	
219	Ø33Ø		1 Ø1Ø 101 ØØØ		MTC	
22Ø	Ø331		1 101 100 Ø10		P,AM1A	IS USERCODE SET TO 6
221	Ø332	Ø336	1 101 111 Ø11		BRN	RET NO, IT IS 7 RETURN TO MANUAL MODE
222	Ø333		1 100 ØØ1 100		PT12	YES, SET TO RUN MODE
223	Ø334		Ø ØØ1 Ø11 ØØØ		LDC1	
224	Ø335		Ø 111 ØØ1 100		PT7	
225	Ø336		Ø ØØØ Ø11 ØØØ	RET	LDCØ	SET USER CODE TO Ø
226	Ø337		Ø Ø1Ø 101 ØØØ		CXM	
227	Ø34Ø		1 100 101 ØØØ		DNR	
228	Ø341		Ø ØØØ 100 ØØØ	DRET	RMGRA	DO A RETURN
229	Ø342	Ø117	Ø 101 ØØØ ØØ1	A1	JSB	RESTOR RESTORE USERCODE
23Ø	Ø343	Ø173	Ø 111 101 111		BRN	A2 GO BACK TO ROM 6
231	Ø344		Ø Ø1Ø Ø1Ø ØØØ	HOME2	ROM 1	GO TO TABLE NUMBER ROUTINE WITHOUT PRINT
232	Ø345		1 110 100 ØØØ		YPOC	CHECK PRINT ON COMMAND FLAG

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	Ø346		1 Ø11 Ø1Ø 1ØØ		YS11		
234	Ø347	Ø362	1 111 ØØ1 Ø11		BRN	HOME1	NOT SET, GO TO PRINT TABLE NUMBER
235	Ø35Ø		1 Ø11 1ØØ 1ØØ		RS11		
236	Ø351	Ø344	1 11Ø Ø1Ø Ø11		BRN	HOME2	SET GO TO TABLE ROUTINE SKIP PRINT
237	Ø352		Ø ØØØ ØØØ ØØØ		DUMMY		
238	Ø353		Ø ØØØ ØØØ ØØØ		DUMMY		
239	Ø354	ØØ17	Ø ØØØ 111 111		BRN	EQUAL	GO TO ROM A EQUAL ROUTINE FROM ROM 4
240	Ø355		1 11Ø 1ØØ ØØØ	PRINTT	YPOC		CHECK PRINT ON COMMAND FLAG
241	Ø356		1 Ø11 Ø1Ø 1ØØ		YS11		
242	Ø357	Ø151	Ø 11Ø 1ØØ 111		BRN	PRINT2	NOT SET, GO TO PINT DATA
243	Ø36Ø		1 Ø11 1ØØ 1ØØ		RS11		
244	Ø361	Ø1Ø6	Ø 1ØØ Ø11 Ø11		BRN	SUPC	SET, GO TO SUPERVISOR
245	Ø362		Ø Ø1Ø Ø1Ø ØØØ	HOME1	ROM 1		GO BACK TO TABLE ROUTINE TO PRINT TABLE NUMBER
246	Ø363		Ø Ø1Ø Ø1Ø ØØØ		ROM 1		CONTINUE DIC CHECK
247	Ø364		Ø ØØØ ØØØ ØØØ		DUMMY		
248	Ø365		1 ØØØ Ø1Ø ØØØ	EEXX	ROM 4		CONTINUE EEX ROUTINE IN ROM 4
249	Ø366		Ø ØØ1 1ØØ Ø1Ø	EEXX1	P,CM1		CHECK IF USERCODE IS OTHER THAN Ø
250	Ø367	Ø371	1 111 1ØØ 111		BRN	EEXX2	YES, GO TO SET MANUAL MODE
251	Ø37Ø	Ø365	1 111 Ø1Ø 111		BRN	EEXX	NO, CONTINUE
252	Ø371		1 1ØØ ØØ1 1ØØ	EEXX2	PT12		
253	Ø372		Ø ØØØ Ø11 ØØØ		LDCØ		SET TO MANUAL MODE
254	Ø373	Ø365	1 111 Ø1Ø 111		BRN	EEXX	CONTONUE
255	Ø374	Ø366	1 111 Ø11 Ø11		BRN	EEXX1	EEX ROUTINE FROM ROM 4
256	Ø375		Ø Ø1Ø 1Ø1 ØØØ	PROGM	CXM		STORE COUNTER FOR 2 DIGIT ENTRY
257	Ø376		1 Ø1Ø ØØ1 1ØØ	PROGM1	PT1Ø		
258	Ø377		Ø ØØ1 Ø11 ØØØ		LDC1		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—1)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000	0160	0 111 000 011		BRN	ENTRY	
4	0001	0053	0 010 101 111		BRN	SIN0	
5	0002	0052	0 010 101 011		BRN	COS0	
6	0003	0054	0 010 110 011		BRN	TAN0	
7	0004	0012	0 000 101 011		BRN	ASIN	
8	0005	0014	0 000 110 011		BRN	ACOS	
9	0006	0016	0 000 111 011		BRN	ATAN	
10	0007	0055	0 010 110 111		BRN	SQT1	
11	0010		0 000 010 000		ROM 0		DATA ENTRY ROUTINE
12	0011	0032	0 001 101 011		BRN	SQUAR	
13	0012		1 010 000 100	ASIN	SS10		
14	0013	0053	0 010 101 111		BRN	SIN0	
15	0014		1 010 000 100	ACOS	SS10		
16	0015	0052	0 010 101 011		BRN	COS0	
17	0016		1 010 000 100	ATAN	SS10		
18	0017	0054	0 010 110 011		BRN	TAN0	
19	0020		0 100 010 000		ROM 2		LOAD PI ROUTINE
20	0021		0 100 010 000		ROM 2		INTEGER X ROUTINE
21	0022	0072	0 011 101 011		BRN	ABS0	ABSOLUTE VALUE ROUTINE
22	0023	0201	1 000 000 111		BRN	POLAR0	TO POLAR ROUTINE
23	0024	0202	1 000 001 011		BRN	RECT0	TO RECTANGULAR ROUTINE
24	0025		1 100 010 000		ROM 6		X EXCHANGE Y ROUTINE
25	0026	0203	1 000 001 111		BRN	ACC+0	ACC+ ROUTINE
26	0027	0220	1 001 000 011		BRN	ACC–	ACC– ROUTINE
27	0030		1 100 010 000		ROM 6		RECALL ACC ROUTINE
28	0031		1 100 010 000		ROM 6		CLEAR ACC ROUTINE
29	0032	0105	0 100 011 001	SQUAR	JSB	SET1	SET RETURN FLAG
30	0033		0 110 001 110		W,CTA		
31	0034		0 000 101 110		W,ZTB		
32	0035	0244	1 010 010 101		JSB	MYP11	
33	0036	0332	1 101 101 011		BRN	OFL1	OVERFLOW TEST AND DISPLAY
34	0037	0345	1 110 010 111		BRN	TABS	FROM PROGRAM MODE IN ROM 6
35	0040		1 010 010 000		ROM 5		DMS TO DWDF
36	0041		1 010 010 000		ROM 5		DWDF TO DMS
37	0042		1 010 010 000		ROM 5		GRAD TO DWDF
38	0043		1 010 010 000		ROM 5		DWDF TO GRAD
39	0044		1 010 010 000		ROM 5		RAD TO DWDF
40	0045		1 010 010 000		ROM 5		DWDF TO RAD
41	0046		1 010 010 000		ROM 5		MILS TO DWDF
42	0047		1 010 010 000		ROM 5		DWDF TO MILS
43	0050	0120	0 101 000 011		BRN	EEX	ENTER EXP ROUTINE
44	0051	0331	1 101 100 111		BRN	ERR1	TABLE NUMBER ERROR
45	0052		1 001 000 100	COS0	SS9		
46	0053		0 101 000 100	SIN0	SS5		
47	0054		0 001 000 100	TAN0	SS1		
48	0055	0223	1 001 001 111	SQT1	BRN	SQT2	
49	0056		0 000 101 110	SQT3	W,ZTB		
50	0057		0 110 010 000		ROM 3		
51	0060	0163	0 111 001 111		BRN	PRXY	PRINT X,Y ROUTINE
52	0061		1 100 010 000		ROM 6		PRINT R,A ROUTINE

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
53	0062	0331	1 101 100 111		BRN	ERR1	TABLE NUMBER ERROR
54	0063	0331	1 101 100 111		BRN	ERR1	TABLE NUMBER ERROR
55	0064	0331	1 101 100 111		BRN	ERR1	
56	0065	0331	1 101 100 111		BRN	ERR1	
57	0066	0331	1 101 100 111		BRN	ERR1	
58	0067	0331	1 101 100 111		BRN	ERR1	
59	0070	0331	1 101 100 111		BRN	ERR1	
60	0071	0331	1 101 100 111		BRN	ERR1	
61	0072		1 101 001 100	ABS0	PT13		ABSOLUTE VALIE ROUTINE
62	0073		0 011 000 010		P,ZTC		CHANGE SIGN TO POSITIVE
63	0074		0 110 001 110		W,CTA		
64	0075	0332	1 101 101 011		BRN	OFL1	
65	0076		1 011 001 110	CHECK0	W,SRA		PROGRAM MODE STORE AND RECALL ROUTINE
66	0077		1 001 001 110		W,SRC		FROM ROM 4
67	0100		0 000 011 100		PRS		
68	0101		0 001 100 100		YP1		
69	0102	0076	0 011 111 011		BRN	CHECK0	SHIFT RIGHT TO PT 0
70	0103	0263	1 011 001 111		BRN	CHECK	CONTINUE WITH MANUAL CHECK
71	0104		0 000 010 000	SUPCA	ROM 0		GO TO SUPERVISOR WITHOUT CHECKING USERCODE
72	0105		0 000 010 000	SUPA1	ROM 0		GO TO SUPERVISOR
73	0106		0 010 101 000	SET1	CXM		SET RETURN FLAG
74	0107		0 000 001 100		PT0		
75	0110		0 001 011 000		LDC1		
76	0111		0 010 101 000		CXM		
77	0112		0 000 110 000		RETURN		
78	0113		0 000 001 100	TAB6	PT0		
79	0114		1 100 101 000		DNR		RESTORE 1ST TABLE NUMBER DIGIT
80	0115		1 110 100 010		P,AXC		COMBINE TWO DIGITS
81	0116		1 110 101 110		W,AXC		PUT TABLE NUMBER IN REGISTER A
82	0117	0333	1 101 101 111		BRN	TAB 2	
83	0120	0276	1 011 111 101	EEX	JSB	ADDR	LOAD DATA STORAGE ADDRESS
84	0121	0266	1 011 011 101		JSB	ADDR0	REGISTER 0 ADDRESS
85	0122		0 010 101 000		CXM		GET M FLAG
86	0123		1 000 010 000		ROM 4		CONTINUE IN ROM 4
87	0124		0 000 000 000		DUMMY		
88	0125		0 000 000 000		DUMMY		
89	0126		0 111 001 100		PT7		SET POINTER FOR USERCODE
90	0127		0 100 011 000		LDC4		LOAD 4 AS USERCODE
91	0130		0 000 011 000		LDC0		LOAD 0 FOR COUNTER
92	0131		0 011 001 100		PT3		
93	0132		0 000 011 000		LDC0		LOAD OP CODE FOR ADD
94	0133		0 001 011 000		LDC1		
95	0134		0 010 101 000		CXM		RESTORE M AND DATA
96	0135	0276	1 011 111 101		JSB	ADDR	LOAD DATA STORAGE ADDRESS
97	0136	0213	1 000 110 001		JSB	ADDR1	DATA STARAGE 1 ADDRESS
98	0137		1 011 110 000		DTDS		STORE ORIGINAL DATA
99	0140	0323	1 101 001 111		BRN	EEX01	GO TO STORE D REGISTER
100	0141		0 011 001 110	EEX0	W,ZTC		CLEAR C
101	0142		0 100 101 000		CTS		STORE 0 IN D
102	0143		1 001 111 010		XS,AM1		CHECK FOR NEGATIVE EXP
103	0144	0164	0 111 010 011		BRN	EEX1	YES, IT IS NEGATIVE,HCONVERT
104	0145		0 001 001 100	EEX5	PT1		NO, IT IS POSITIVE
105	0146		0 100 001 110	EEX2	W,SLA		NORMALIZE EXPONENT
106	0147		0 000 111 100		PLS		
107	0150		1 100 101 100		YP12		
108	0151	0146	0 110 011 011		BRN	EEX2	
109	0152		1 001 100 010		P,AM1		CHECK FOR LEADING ZERO
110	0153	0210	1 000 100 011		BRN	EEX3	NO ZERO
111	0154		0 100 010 010		WP,SLA		ZERO, SHIFT LEFT ONE
112	0155		1 010 101 000	EEX4	MTC		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156		1 110 101 110		W,AXC		DATA IN C, M IN A	
114	0157	0105	0 100 010 111		BRN	SUPA1	GO TO SUPERVISOR	
115	0160		0 000 010 000	ENTRY	ROM 0		GO TO ROM 0 TO SAVE OLD USERCODE	
116	0161		0 011 100 100		RS3		RETURN FROM ROM 0 RESET 3	
117	0162	0172	0 111 101 011		BRN	ENTRY1		
118	0163		1 100 010 000	PRXY	ROM 6		PRINT X,Y ROUTINE IN ROM 6	
119	0164		1 110 101 110	EEX1	W,AXC		PUT DATA IN C, 0 IN A	
120	0165		0 101 001 010		X,AMCC		COMPLIMENT EXP	
121	0166		0 010 001 100		PT2			
122	0167		1 001 011 000		LDC9		LOAD NEGATIVE SIGN	
123	0170		1 110 101 110		W,AXC		PUT EXP IN A	
124	0171	0146	0 110 011 011		BRN	EEX2	CONTINUE	
125	0172		0 101 011 000	ENTRY1	LDC5		SET DIC TO 5	
126	0173		1 001 011 000		LDC9		SET ALIGNMENT SHIFT TO 9	
127	0174		0 111 001 100		PT7			
128	0175		0 011 011 000		LDC3		SET USERCODE TO 3 FOR POSSIBLE CANCEL	
129	0176		0 010 101 000		CXM			
130	0177		1 110 010 000	PRINT1	ROM 7		GO TO PRINT "ENTER ANGLE"	
131	0200		1 110 010 000	PRINT	ROM 7			
132	0201		1 100 010 000	POLAR0	ROM 6		TO POLAR ROUTINE IN ROM 6	
133	0202		1 100 010 000	RECT0	ROM 6		TO RECT ROUTINE IN ROM 6	
134	0203		1 100 010 000	ACC+0	ROM 6		ACCUMULATE + ROUTINE IN ROM 6	
135	0204		1 100 010 000	ACC–0	ROM 6		ACCUMULATE – ROUTINE IN ROM 6	
136	0205		1 101 100 010	CHECK1	P,AM1A		IS 1ST DIGIT 5	
137	0206	0331	1 101 100 111		BRN	ERR1	NO, TABLE NUMBER ERROR	
138	0207	0216	1 000 111 011		BRN	RECAL	YES, GO TO RECALL ROUTINE	
139	0210		1 111 101 010	EEX3	X,AP1A		ADD 1 TO EXP OF NORMALIZED EXP	
140	0211	0155	0 110 110 111		BRN	EEX4	CONTINUE	
141	0212		0 100 011 000	ADDR4	LDC4			
142	0213	0270	1 011 100 011		BRN	ADDRC	LOAD DATA STORAGE 4 ADDRESS	
143	0214		0 001 011 000	ADDR1	LDC1		LOAD DATASTORAGE 1 ADDRESS	
144	0215	0270	1 011 100 011		BRN	ADDRC	CONTINUE	
145	0216		0 100 010 000	RECAL	ROM 2		GO TO RECALL ROUTINE	
146	0217		0 100 010 000	STORE	ROM 2		GO TO STORE ROUTINE	
147	0220		1 000 100 100	ACC–	RS8			
148	0221	0204	1 000 010 011		BRN	ACC–0		
149	0222		0 000 110 000		RETURN			
150	0223		0 000 001 100	SQT2	PT0			
151	0224		0 010 101 000		CXM			
152	0225		0 011 011 000		LDC3		SET TRIG AND SQRT RETURN FLAG	
153	0226		0 010 101 000		CXM			
154	0227		1 100 001 100		PT12			
155	0230		0 110 001 110		W,CTA			
156	0231	0056	0 010 111 011		BRN	SQT3		
157	0232		0 100 010 000	TAB1	ROM 2			
158	0233		0 000 101 110	ADD3	W,ZTB		ADDITION ROUTINE	
159	0234		1 111 111 010		XS,AP1A			
160	0235		1 111 111 010		XS,AP1A			
161	0236		0 111 111 010		XS,CP1C			
162	0237		0 111 111 010		XS,CP1C			
163	0240		0 001 001 010		X,AMC			
164	0241	0243	1 010 001 111		BRN	ADD4		
165	0242		1 110 101 110		W,AXC			
166	0243		1 110 100 110	ADD4	M,AXC			
167	0244	0250	1 010 100 011		BRN	ADDD		
168	0245		1 000 010 000	MPY11	ROM 4			
169	0246		0 101 001 010	DIV11	X,AMCC			
170	0247		1 000 010 000		ROM 4			
171	0250		0 110 100 110	ADDD	M,ZMC			
172	0251	0253	1 010 101 111		BRN	ADD5		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		1 110 101 110		W,AXC		
174	0253		1 000 100 110	ADD5	M,BXC		
175	0254		0 001 001 010	ADD6	X,AMC		
176	0255	0276	1 011 111 011		BRN	ADD7	
177	0256		1 010 001 110		W,SRB		
178	0257		1 111 101 010		X,AP1A		
179	0260		0 000 001 110		W,ZMB		
180	0261	0276	1 011 111 011		BRN	ADD7	
181	0262	0254	1 010 110 011		BRN	ADD6	
182	0263		1 101 000 010	CHECK	P,AMCA		CHECK WHETHER 1ST TAB DIGIT IS 4
183	0264		1 101 100 010		P,AM1A		
184	0265	0205	1 000 010 111		BRN	CHECK1	NO, CONTINUE CHECK
185	0266	0217	1 000 111 111		BRN	STORE	YES, GO TO STORE ROUTINE
186	0267		0 000 011 000	ADDR0	LDC0		
187	0270		1 001 110 000	ADDRC	ATDS		
188	0271		0 010 001 110		W,BTC		
189	0272		0 000 110 000		RETURN		
190	0273	0320	1 101 000 011		BRN	ERROR	
191	0274		1 110 101 110		W,AXC		PUT DATA IN A
192	0275	0305	1 100 010 111		BRN	EEX9	CONTINUE EXP ROUTINE
193	0276		0 110 010 000	ADD7	ROM 3		
194	0277		1 000 101 110	ADDR	W,BXC		LOAD DS ADDRESS ROUTINE
195	0300		0 011 001 110		W,ZTC		
196	0301		0 111 101 110		W,CP1C		
197	0302		1 100 001 100		PT12		
198	0303		1 010 011 000		LDC10		
199	0304		0 000 110 000		RETURN		
200	0305	0276	1 011 111 101	EEX9	JSB	ADDR	
201	0306	0211	1 000 101 001		JSB	ADDR4	LOAD DATA STORAGE 4 ADDRESS
202	0307		1 011 111 000		DSTC		RELOAD OLD D REGISTER
203	0310		0 100 101 000		CTS		RESTORE OLD D REGISTER
204	0311	0276	1 011 111 101		JSB	ADDR	
205	0312	0266	1 011 011 101		JSB	ADDR0	LOAD DATA STORAGE ADDRESS
206	0313		1 011 111 000		DSTC		RELOAD ORIGINAL M VECTOR
207	0314		0 010 101 000		CXM		REPLACE M
208	0315	0276	1 011 111 101		JSB	ADDR	
209	0316	0213	1 000 110 001		JSB	ADDR1	LOAD DATA STORAGE 1 ADDRESS
210	0317	0376	1 111 111 011		BRN	EEX10	GO TO ROM 4 TO FINISH ROUTINE
211	0320	0276	1 011 111 101	ERROR	JSB	ADDR	
212	0321	0266	1 011 011 101		JSB	ADDR0	LOAD DATA STORAGE SDDRESS
213	0322	0377	1 111 111 111		BRN	ERROR2	
214	0323		0 110 101 000	EEX01	STA		LOAD D REGISTER TO A
215	0324		1 110 101 110		W,AXC		PUT IT IN C
216	0325	0276	1 011 111 101		JSB	ADDR	
217	0326	0211	1 000 101 001		JSB	ADDR4	LOAD DATA STORAGE 4 ADDRESS
218	0327		1 011 110 000		DTDS		STORE D IN DS 2
219	0330	0141	0 110 000 111		BRN	EEX0	CONTINUE
220	0331		1 110 010 000	ERR1	ROM 7		
221	0332		0 100 010 000	OFL1	ROM 2		
222	0333		0 001 001 100	TAB2	PT1		
223	0334		0 100 011 000		LDC4		LOAD 4 TO CHECK TABLE NUMBER
224	0335		0 001 001 100		PT1		
225	0336		0 001 000 010		P,AMC		IS FIRST DIGIT GREATER THAN 4
226	0337	0263	1 011 001 111		BRN	CHECK	YES, CHECK WHETHER ITS STORE OR RECALL
227	0340		0 100 001 110	TAB3	W,SLA		SHIFT LEFT ONE
228	0341		0 000 111 100		PLS		INCREMENT COUNTER
229	0342		1 101 101 100		YP13		IS COUNTER 13
230	0343	0340	1 110 000 011		BRN	TAB3	NO, CONTINUE SHIFT
231	0344		0 000 010 000		ROM 0		GO TO CHECK PRINT ON COMMAND FLAG
232	0345		1 110 101 110	TAB5	W,AXC		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 111 110 000		CTT		LOAD TO C AND T
234	0347		0 101 000 000		IS2		
235	0350		0 100 101 000		IXT		EXCHANGE T WITH I/O REGISTER
236	0351		1 001 000 000		IS1		
237	0352		0 010 001 110		W,BTC		
238	0353		1 010 100 000		SBL		SET BUSY LIGHT
239	0354		0 001 010 000		EERA		EXTERNAL ROM ADDRESS ENTRY
240	0355		0 100 001 110	TAB4	W,SLA		SHIFT LEFT ONE
241	0356		0 010 101 000		CXM		RESTORE FLAG VECTOR
242	0357		1 110 101 110		W,AXC		
243	0360		0 100 101 000		CTS		STORE FIRST DIGIT
244	0361		0 010 001 110		W,BTC		RESTORE C
245	0362	0105	0 100 010 111		BRN	SUPA1	
246	0363	0200	1 000 000 011		BRN	PRINT	GO TO PRINT TABLE NUMBER FROM ROM 0
247	0364		0 101 100 010		P,CM1C		IS DIC 4
248	0365	0232	1 001 101 011		BRN	TAB1	NO,CONTINUE CHECK
249	0366		1 010 001 100		PT10		
250	0367		0 010 101 000		CXM		GET COUNTER IN FLAG VECTOR
251	0370		0 101 100 010		P,CM1C		DECREASE COUNTER
252	0371	0355	1 110 110 111		BRN	TAB4	COUNTER GREATER THAN 0?
253	0372		1 011 001 100		PT11		NO,SET DIC TO 0
254	0373		0 000 011 000		LDC0		
255	0374		0 010 101 000		CXM		
256	0375	0113	0 100 101 111		BRN	TAB6	
257	0376		1 000 010 000	EEX10	ROM 4		GO TO ROM 4 TOFINISH ROUTINE
258	0377		1 000 010 000	ERROR2	ROM 4		GO TO ERROR ROUTINE

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—2)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000	0002	0 000 001 011		BRN	DISP2	NO, CONTINUE
4	0001		1 001 011 000		LDC9		YES, BLANK IT
5	0002		0 011 001 100	DISP2	PT3		BLANK DIGITS 2 AND 3
6	0003		1 001 011 000		LDC9		
7	0004		0 000 011 000		LDC0		
8	0005		1 001 100 010		P,AM1		IS DIGIT U ZERO
9	0006	0010	0 000 100 011		BRN	DISP3	NO, CONTINUE
10	0007		1 001 011 000		LDC9		YES, BLANK IT
11	0010		1 000 101 110	DISP3	W,BXC		RESTORE DATA
12	0011		0 000 101 000		DSTO		DISPLAY ON
13	0012		0 000 110 000		RETURN		
14	0013		1 011 110 000	STORE3	DTDS		
15	0014	0056	0 010 111 011		BRN	OPUT	
16	0015	0027	0 001 011 111		BRN	OFLL0	
17	0016		1 010 010 000	DTRD	ROM 5		GO BACK TO ANGLE CONVERSION
18	0017		1 011 111 000	RECAL1	DSTC		LOAD DATA TO C
19	0020	0056	0 010 111 011		BRN	OPUT	GO TO OUTPUT
20	0021	0164	0 111 010 011		BRN	PI1	LOAD PI ROUTINE
21	0022	0166	0 111 011 011		BRN	INT1	INTEGER ROUTINE
22	0023		0 000 110 000		RETURN		
23	0024		0 011 010 000	OFL0	YS3		CHECK FOR LOAD PI FLAG
24	0025	0106	0 100 011 011		BRN	OFL01	
25	0026	0016	0 000 111 011		BRN	DTRD	
26	0027		0 000 110 100	OFLL0	CLS		
27	0030		0 111 000 100		SS7		
28	0031		0 001 100 110	OFLL1	M,CM1		MANTISSA ZERO?
29	0032	0034	0 001 110 011		BRN	OFL2	NO, CONTINUE TEST
30	0033	0045	0 010 010 111		BRN	OFL4	YES, SET RESULT TO ZERO
31	0034		0 110 111 010	OFL2	XS,ZMC		CHECK SIGN OF EXPONENT
32	0035	0105	0 100 010 111		BRN	SUPA20	RESULT OK
33	0036		0 010 101 010		X,ZMCC		SIGN CODE DECREMENT
34	0037		0 101 111 010		XS,CM1C		CONTINUE TEST, EXPONENT COMPLEMENTED
35	0040	0043	0 010 001 111		BRN	OFL3	NO CARRY, TROUBLE
36	0041		1 110 101 110		W,AXC		CARRY, RESULT OK
37	0042	0105	0 100 010 111		BRN	SUPA20	
38	0043		0 001 111 010	OFL3	XS,CM1		
39	0044	0047	0 010 011 111		BRN	OFL5	NO CARRY, OVERFLOW
40	0045		0 011 001 110	OFL4	W,ZTC		CARRY, UNDERFLOW, RESULT SET TO ZERO
41	0046	0105	0 100 010 111		BRN	SUPA20	GO TO SUPERVISOR
42	0047		0 011 001 110	OFL5	W,ZTC		
43	0050		0 101 101 110		W,CM1C		SET ALL 9'S
44	0051		0 010 001 100		PT2		
45	0052		0 000 011 000		LDC0		SET EXPONENT SIGN TO POSITIVE
46	0053		1 101 001 100		PT13		
47	0054		0 000 011 000		LDC0		SET MANTISSA SIGN TO POSITIVE
48	0055	0102	0 100 001 011		BRN	PRINT1	PRINT ERROR MESSAGE FOR OVERFLOW
49	0056		0 010 101 000	OPUT	CXM		OUTPUT SYBOL FOR STORE AND RECALL
50	0057	0156	0 110 111 011		BRN	OPUT1	GO TO CHECK OP MODE
51	0060		1 011 111 110	NRM21	S,ZTA		
52	0061		1 100 001 100		PT12		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062		0 000 101 110		W,ZTB		
54	0063		1 001 100 010	NRM23	P,AM1		
55	0064	0072	0 011 101 011		BRN	NRM24	
56	0065		0 100 001 110		W,SLA		
57	0066		0 101 101 010		X,CM1C		
58	0067		1 001 101 110		W,AM1		
59	0070	0063	0 011 001 111		BRN	NRM23	
60	0071		0 011 001 110		W,ZTC		
61	0072		0 100 101 010	NRM24	X,ATB		
62	0073		1 110 001 110		W,APBA		
63	0074		1 001 111 110		S,AM1		
64	0075	0225	1 001 010 111		BRN	MPY28	
65	0076		1 110 100 110		M,AXC		
66	0077		0 110 001 110	NRM25	W,CTA		
67	0100		0 000 101 110		W,ZTB		
68	0101		1 000 010 000	NRM26	ROM 4		
69	0102		1 110 010 000	PRINT1	ROM 7		PRINT OVERFLOW MESSAGE
70	0103		0 001 100 100	SQRT1	RS1		
71	0104	0216	1 000 111 011		BRN	SQRT	RETURN AFTER SQRT
72	0105		0 000 010 000	SUPA20	ROM 0		
73	0106		0 010 010 100	OFL01	YS2		
74	0107	0027	0 001 011 111		BRN	OFLL0	
75	0110		0 010 100 100		RS2		
76	0111		0 000 001 100		PT0		
77	0112		0 010 101 000		CXM		
78	0113		0 110 011 000		LDC6		
79	0114		0 010 101 000		CXM		
80	0115		1 001 010 100		YS9		
81	0116	0121	0 101 000 111		BRN	A0	
82	0117		1 001 100 100		RS9		
83	0120	0214	1 000 110 011		BRN	COS	
84	0121		0 101 010 100	A0	YS5		
85	0122	0125	0 101 010 111		BRN	A1	
86	0123		0 101 100 100		RS5		
87	0124	0213	1 000 101 111		BRN	SIN	
88	0125		1 010 010 100	A1	YS10		
89	0126	0103	0 100 001 111		BRN	SQRT1	
90	0127		1 010 100 100		RS10		
91	0130	0215	1 000 110 111		BRN	ATAN1	
92	0131		0 110 100 100	RECAL	RS6		
93	0132	0134	0 101 110 011		BRN	STORE1	
94	0133		0 110 000 100	STORE	SS6		
95	0134		0 000 001 100	STORE1	PT0		
96	0135		0 100 001 110	STORE0	W,SLA		
97	0136		0 000 111 100		PLS		SHIFT ENTER DS REGISTER NUMBER TO PT11
98	0137		1 011 101 100		YP11		
99	0140	0135	0 101 110 111		BRN	STORE0	
100	0141		0 011 001 110		W,ZTC		
101	0142		1 100 001 100		PT12		LOAD DS ADDRESS
102	0143		0 111 101 110		W,CP1C		
103	0144		1 010 011 000		LDC10		
104	0145	0150	0 110 100 011		BRN	STORE2	
105	0146		0 000 000 000		DUMMY		
106	0147		0 000 010 000	SUPBA	ROM 0		GO TO SUPERVISOR A WITHOUT DISPLAY
107	0150		1 110 100 010	STORE2	P,AXC		LOAD DS REGISTER ADDRESS
108	0151		1 001 110 000		ATDS		
109	0152		1 000 101 110		W,BXC		YES, RESTORE C SAVE DS ADDRESS
110	0153		0 110 010 100		YS6		
111	0154	0017	0 000 111 111		BRN	RECAL1	NO, GO TO RECALL
112	0155	0013	0 000 101 111		BRN	STORE	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		1 100 001 100	OPUT1	PT12		
114	0157		0 001 100 010		P,CM1		PROGRAM MODE?
115	0160	0223	1 001 001 111		BRN	SUPA4	YES, GO TO SUPERVISOR
116	0161	0221	1 001 000 111		BRN	OPUT2	NO, GO TO PRINT SYMBOL
117	0162		0 000 000 000		DUMMY		
118	0163		0 000 010 000	PRINT3	ROM 0		GO TO RESTORE USERCODE IN ROM 0
119	0164		0 001 100 100	PI1	RS1		
120	0165		0 110 010 000		ROM 3		LOAD PI IN ROM 3
121	0166		0 110 001 110	INT1	W,CTA		STORE ORIGINAL DATA
122	0167		0 010 001 100		PT2		
123	0170		0 001 100 010		P,CM1		CHECK WHETHER EXPONENT IS NEGATIVE
124	0171	0211	1 000 100 111		BRN	INT2	YES, SET VALUE TO ZERO
125	0172		0 001 001 100		PT1		NO, CHECK IF EXPONENT IS GREATER THAN 10
126	0173		0 001 100 010		P,CM1		
127	0174	0105	0 100 010 111		BRN	SUPA20	YES, DO NOTHING
128	0175		0 000 001 100		PT0		CHECK IF EXPONENT IS LESS THAN 10
129	0176		0 001 100 010		P,CM1		BUT GREATER THAN ZERO
130	0177	0202	1 000 001 011		BRN	INT4	YES, PREPARE TO ZERO DIGITS
131	0200		1 011 001 100		PT11		NO, EXPONENT IS ZERO
132	0201	0206	1 000 011 011		BRN	INT6	ZERO ALL DIGITS EXCEPT THE FIRST
133	0202		1 100 001 100	INT4	PT12		
134	0203		0 000 011 100	INT5	PRS		SET POINTER
135	0204		0 101 101 010		X,CM1C		ZERO DIGIT AFTER DECIMAL POINT
136	0205		1 000 001 111		BRN	INT5	
137	0206		0 011 010 010	INT6	WP,ZTC		
138	0207		1 110 101 010		X,AXC		RESTORE ORIGINAL EXPONENT
139	0210	0105	0 100 010 111		BRN	SUPA20	
140	0211		0 011 001 110	INT2	W,ZTC		
141	0212	0105	0 100 010 111		BRN	SUPA20	
142	0213		1 100 010 000	SIN	ROM 6		RETURN IN ROM 6
143	0214		1 100 010 000	COS	ROM 6		RETURN IN ROM 6
144	0215		1 100 010 000	ATAN1	ROM 6		RETURN IN ROM 6
145	0216		1 100 010 000	SQRT	ROM 6		RETURN IN ROM 6
146	0217	0131	0 101 100 111		BRN	RECAL	NO, RECAL
147	0220	0133	0 101 101 111		BRN	STORE	YES, STORE
148	0221		0 010 101 000	OPUT2	CXM		
149	0222	0245	1 010 010 111		BRN	OPUT3	GO TO PRINT SYMBOL IN ROM 7
150	0223		0 010 101 000	SUPA4	CXM		
151	0224	0227	1 001 011 111		BRN	SUPA41	
152	0225		1 000 010 000	MPY28	ROM 4		
153	0226	0060	0 011 000 011		BRN	NRM21	
154	0227		0 000 110 100	SUPA41	CLS		
155	0230		0 111 000 100		SS7		
156	0231	0105	0 100 010 111		BRN	SUPA20	GO TO SUPERVISOR
157	0232		0 010 010 000	ADD31	ROM 1		GO TO ADD ROUTINE
158	0233		0 101 100 010		P,CM1C		IS DIC 5
159	0234	0105	0 100 010 111		BRN	SUPA20	
160	0235	0247	1 010 011 111		BRN	ENTRY	
161	0236		0 010 101 000	PRINT2	CXM		
162	0237		1 000 101 110		W,BXC		
163	0240		1 010 101 000		MTC		
164	0241		1 110 101 110		W,AXC		PUT M IN A FOR RESTORE ROUTINE
165	0242		1 000 101 110		W,BXC		RESTORE C
166	0243	0163	0 111 001 111		BRN	PRINT3	
167	0244		0 000 000 000		DUMMY		
168	0245		1 000 101 110	OPUT3	W,BXC		PUT DS ADDRESS IN C, DATA IN B
169	0246		1 110 010 000		ROM 7		PRINT SYMBOL AND DATA IN ROM 7
170	0247		0 010 001 110	ENTRY	W,BTC		
171	0250		0 000 001 100		PT0		
172	0251		1 110 100 010		P,AXC		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 101 010 100	YS5		IS D.P. SET	
174	0253	0303	1 100 001 111	BRN	ENTRY2	NO, ENTER DIGIT	
175	0254		0 010 101 000	CXM		YES, GET FLAT VECTOR	
176	0255		0 110 001 110	W,CTA		PREPARE TO CHECK IF M10 IS 9	
177	0256		1 010 001 100	PT10			
178	0257		0 110 011 000	LDC6			
179	0260		1 010 001 100	PT10			
180	0261		0 001 000 010	P,AMC			
181	0262	0273	1 011 101 111	BRN	ENTRY3	M10 IS 9	
182	0263		0 010 011 000	LDC2		NO, PREPARE TO CHECK IF M10 IS 5	
183	0264		1 010 001 100	PT10			
184	0265		0 001 000 010	P,AMC			
185	0266	0301	1 100 000 111	BRN	ENTRY 4	M10 IS 5	
186	0267		1 011 001 100	PT11		NO	
187	0270		0 000 011 000	LDC0		SET DIC TO 0	
188	0271		0 111 000 100	SS7			
189	0272	0236	1 001 111 011	BRN	PRINT2		
190	0273		0 101 011 000	ENTRY3 LDC5		SET M10 TO 5, ONE DECIMAL ENTERED	
191	0274		0 010 101 000	ENTRY5 CXM		RESTORE FLAG VECTOR	
192	0275		0 110 001 110	W,CTA			
193	0276		0 101 100 100	RS5		RESET D.P. FLAG	
194	0277	0352	1 110 101 101	ENTR12 JSB	DISP0	GO TO DISPLAY	
195	0300	0147	0 110 011 111	BRN	SUPBA		
196	0301		0 001 011 000	ENTRY4 LDC1		SET M10 TO 1, TWO DECIMAL ENTERED	
197	0302	0274	1 011 110 011	BRN	ENTRY5		
198	0303		1 000 101 110	ENTRY2 W,BXC			
199	0304		1 010 101 000	MTC			
200	0305		0 110 001 110	W,CTA			
201	0306		1 010 001 100	PT10		PREPARE TO CHECK IF M10 IS 9	
202	0307		0 110 011 000	LDC6			
203	0310		1 010 001 100	PT10			
204	0311		0 001 000 010	P,AMC			
205	0312	0322	1 101 001 011	BRN	ENTRY7	M10 IS 9	
206	0313		0 010 011 000	LDC2		PREPARE TO CHECK IF M10 IS 5	
207	0314		1 010 001 100	PT10			
208	0315		0 001 000 010	P,AMC			
209	0316	0325	1 101 010 111	BRN	ENTRY8	M10 IS 5	
210	0317		1 100 101 110	W,AXB		RESTORE ENTERED DATA	
211	0320		0 010 001 100	PT2		SET POINTER TO 2	
212	0321	0327	1 101 011 111	BRN	ENTRY9	GO TO SHIFT ROUTINE	
213	0322		1 100 101 110	ENTRY7 W,AXB		RESTORE ENTERED DATA	
214	0323		1 100 001 100	PT12		SET POINTER TO 12	
215	0324	0327	1 101 011 111	BRN	ENTRY9	GO TO SHIFT ROUTINE	
216	0325		1 100 101 110	ENTRY8 W,AXB		RESTORE ENTERED DATA	
217	0326		0 110 001 100	PT6		SET POINTER TO 6	
218	0327		0 100 010 010	ENTRY9 WP,SLA		SHIFT LEFT ONE	
219	0330		1 100 101 100	YP12			
220	0331	0336	1 101 111 011	BRN	ENTR10	NO, CONTINUE CHECK	
221	0332	0334	1 101 110 011	BRN	ENTR		
222	0333	0024	0 001 010 011	BRN	OFL0		
223	0334		1 001 001 100	ENTR PT9			
224	0335	0342	1 110 001 011	BRN	ENTR11	SHIFT LEFT 8 PLACES	
225	0336		0 110 101 100	ENTR10 YP6		IS POINTER 6	
226	0337	0277	1 011 111 111	BRN	ENTR12	NO, GO TO DISPLAY ROUTINE	
227	0340		0 101 001 100	PT5		YES, SET POINTER TO 5	
228	0341	0346	1 110 011 011	BRN	ENTR13	SHIFT LEFT 4 PLACES	
229	0342		0 100 010 010	ENTR11 WP,SLA			
230	0343		0 100 010 010	WP,SLA			
231	0344		0 100 010 010	WP,SLA			
232	0345		0 100 010 010	WP,SLA			

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
233	0346		0 100 010 010	ENTR13	WP,SLA	
234	0347		0 100 010 010		WP,SLA	
235	0350		0 100 010 010		WP,SLA	
236	0351		0 100 010 010		WP,SLA	
237	0352	0277	1 011 111 111		BRN	ENTR12
238	0353		1 110 101 110	DISP0	W,AXC	
239	0354		0 110 001 110		W;CTA	
240	0355		1 011 001 110		W,SRA	SHIFT RIGHT FOR DISPLAY FORMAT
241	0356		1 000 101 110		W,BXC	STORE C
242	0357		0 011 001 110		W,ZTC	PREPARE MASK
243	0360		1 100 001 100		PT12	BLANK DIGIT 12
244	0361		1 001 011 000		LDC9	
245	0362		0 000 000 000		NOP	
246	0363		1 001 100 010		P,AM1	IS DIGIT 11 ZERO?
247	0364	0374	1 111 110 011		BRN	DISP1 NO, CONTINUE
248	0365		1 001 011 000		LDC9	YES, BLANK IT
249	0366		1 001 100 010		P,AM1	IS DIGIT 10 ZERO
250	0367	0374	1 111 110 011		BRN	DISP1 NO, CONTINUE
251	0370		1 001 011 000		LDC9	YES, BLANK IT
252	0371		1 001 100 010		P,AM1	IS DIGIT 9 ZERO
253	0372	0374	1 111 110 011		BRN	DISP1 NO, CONTINUE
254	0373		1 001 011 000		LDC9	YES, BLANK IT
255	0374		0 111 001 100	DISP1	PT7	BLANK DIGIT 6 AND 7
256	0375		1 001 011 000		LDC9	
257	0376		1 001 011 000		LDC9	
258	0377		1 001 100 010		P,AM1	IS DIGIT #5 ZERO

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—3)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000	0363	1 111 001 111		BRN	TAN13	
4	0001		1 100 101 110	TAN15	W,AXB		
5	0002	0047	0 010 100 001		JSB	TNM11	NORMALIZE DATA
6	0003		0 110 101 000		STA		
7	0004	0047	0 010 100 001		JSB	TNM11	
8	0005		0 110 101 000		STA		
9	0006		1 001 010 100		YS9		SEE WHETHER COS IS ASKED
10	0007	0011	0 000 100 111		BRN	TAN16	
11	0010		1 110 101 110		W,AXC		
12	0011		0 101 010 100	TAN16	YS5		
13	0012	0022	0 001 001 011		BRN	ASN12	FORM TAN BY DIVISION OR
14	0013		0 011 011 110		S,ZTC		PREPARE TO FORM SIN OR COS
15	0014	0245	1 010 011 001		JSB	DIV11	
16	0015		0 100 101 000	ASN11	CTS		CALCULATE STN OR COS FROM
17	0016	0244	1 010 010 101		JSB	MPY11	TANGENT, OR FORM ARC TAN
18	0017	0227	1 001 100 001		JSB	ADD10	FROM ARC SIN RO ARCCOS
19	0020	0044	0 010 010 101		JSB	SQT11	
20	0021		0 110 101 000		STA		
21	0022	0245	1 010 011 001	ASN12	JSB	DIV11	
22	0023		1 010 010 100		YS10		
23	0024	0332	1 101 101 011		BRN	RTN12	SIN RO COS OUTPUT
24	0025		1 011 101 110	ATN11	W,ZTA		ARC TAN ROUTINE
25	0026		1 111 100 010		P,AP1A		ATN11 TO ATN13 PREPARE DATA
26	0027		0 100 100 110		M,ATB		FOR FINDING MULTIPLIERS OF
27	0030		1 110 100 110		M,AXC		TAN 0 WHERE TAN 0=10↑–N
28	0031		0 101 101 010	ATN12	X,CM1C		N FROM 0 TO 4
29	0032		1 010 010 010		WP,SRB		
30	0033		0 110 111 010		XS,ZMC		
31	0034	0031	0 001 100 111		BRN	ATN12	
32	0035		1 011 010 010	ATN13	WP,SRA		Y VALUE IS SET TO 1
33	0036		0 111 101 010		X,CP1C		X VALUE THE DATA VALUE
34	0037	0035	0 001 110 111		BRN	ATN13	
35	0040		1 011 001 110		W,SRA		
36	0041		1 010 001 110		W,SRB		
37	0042		0 100 101 000		CTS		
38	0043		1 000 101 110	ATN14	W,BXC		
39	0044	0101	0 100 000 111		BRN	ATN18	
40	0045		1 000 101 110	SQT11	W,BXC		
41	0046		0 100 001 100		PT4		
42	0047	0336	1 101 111 011		BRN	SQT14	
43	0050		0 100 101 000	TNM11	CTS		
44	0051		1 110 101 110		W,AXC		
45	0052		0 110 100 010		P,ZMC		
46	0053	0055	0 010 110 111		BRN	TNM12	
47	0054		0 010 101 110		W,ZMCC		
48	0055		0 110 001 110	TNM12	W,CTA		
49	0056		0 010 001 010		X,BTC		
50	0057	0313	1 100 101 111		BRN	ADD15	
51	0060		0 110 001 110	SIN11	W,CTA		
52	0061		0 001 010 100		YS1		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0062	0045	0 010 010 111		BRN	SQT11		
54	0063		1 010 010 100		YS10			
55	0064	0155	0 110 110 111		BRN	TAN12		
56	0065		0 101 010 100		YS5			
57	0066	0025	0 001 010 111		BRN	ATN11		
58	0067		0 011 111 110		S,ZNCC			
59	0070		1 110 111 110		S,AXC			
60	0071	0015	0 000 110 111		BRN	ASN11		
61	0072		1 010 010 010	ATN15	WP,SRB			
62	0073		1 101 111 110	ATN16	S,AM1A			
63	0074	0072	0 011 101 011		BRN	ATN15		
64	0075		0 111 111 110		S,CP1C			
65	0076		1 100 110 010		WP,AXB			
66	0077		0 111 010 010		WP,APCC			
67	0100		1 100 101 110		W,AXB			
68	0101		0 100 101 110	ATN18	W,ATB			ATN14 TO ATN18 ACTUALLY
69	0102		1 101 010 010		WP,AMCA			FINDS THE MULTIPLIER
70	0103	0073	0 011 101 111		BRN	ATN16		
71	0104		0 110 101 000		STA			
72	0105		1 011 001 110		W,SRA			
73	0106		1 110 110 010		WP,AXC			
74	0107		1 100 101 110		W,AXB			
75	0110		0 100 010 010		WP,SLA			
76	0111		0 100 101 000		CTS			
77	0112		1 111 111 110		S,AP1A			
78	0113		1 111 111 110		S,AP1A			
79	0114	0043	0 010 001 111		BRN	ATN14		
80	0115		0 011 001 110		W,ZTC			
81	0116		0 000 101 010		X,ZTB			
82	0117		1 011 010 110		MS,SRA			
83	0120	0261	1 011 001 001		JSB	DIV14		
84	0121		0 101 100 010		P,CM1C			
85	0122		0 110 101 000		STA			
86	0123		1 110 101 110		W,AXC			
87	0124		0 100 001 100		PT4			
88	0125	0243	1 010 010 001	ATN17	JSB	PQO13		
89	0126		0 110 001 100		PT6			FORM TAN10↑–4 X FIND PSEUDO MULTIPLIER
90	0127	0232	1 001 101 101		JSB	PMU11		
91	0130		1 000 001 100		PT8			
92	0131	0232	1 001 101 101		JSB	PMU11		FORM TAN10↑–3 X FIND PSEUDO MULTIPLIER
93	0132		0 010 001 100		PT2			
94	0133		1 000 011 000		LDC8			
95	0134		1 010 001 100		PT10			
96	0135	0232	1 001 101 101		JSB	PMU11		FORM TAN 10↑–2 X FIND PSEUDO MULTIPLIER
97	0136	0215	1 000 111 001		JSB	ATCD1		
98	0137	0232	1 001 101 101		JSB	PMU11		FORM TAN 10↑–1 X FIND PSEUDO MULTIPLIER
99	0140	0313	1 100 110 001		JSB	ATC1		LOAD PI/4=TAN1 X FORM
100	0141		0 100 001 110		W,SLA			PSEUDO MULTIPLIER
101	0142	0232	1 001 101 101		JSB	PMU11		
102	0143		0 010 001 110		W,BTC			
103	0144	0312	1 100 101 101		JSB	ADD15		
104	0145	0313	1 100 110 001		JSB	ATC1		
105	0146		1 010 101 110		W,CPCC			CONVERSION FROM RADIAN
106	0147	0245	1 010 011 001		JSB	DIV11		TO DEGREE
107	0150		1 001 010 100		YS9			
108	0151	0154	0 110 110 011		BRN	ATN19		
109	0152		0 011 111 110		S,ZNCC			
110	0153	0227	1 001 100 001		JSB	ADD10		
111	0154		0 001 100 100	ATN19	RSI			
112	0155		0 011 001 110	TAN12	W,ZTC			BEGINNING OF TANGENT

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 101 100 010		P,CM1C		ROUTINE, CONVERSION
114	0157		0 111 101 010		X,CP1C		OF DEGREE TO RADIAN
115	0160		0 001 010 100		YS1		
116	0161	0245	1 010 010 111		BRN	MPY11	
117	0162	0245	1 010 011 001		JSB	DIV11	
118	0163	0313	1 100 110 001		JSB	ATC1	
119	0164		1 010 101 110		W,CPCC		
120	0165	0244	1 010 010 101		JSB	MPY11	
121	0166	0313	1 100 110 001	LPI11	JSB	ATC1	LOAD PI/4
122	0167		1 010 101 110		W,CPCC		
123	0170		1 010 101 110		W,CPCC		PI
124	0171	0224	1 001 010 101		JSB	RTN11	
125	0172		1 010 101 110		W,CPCC		LOAD 2 PI
126	0173	0352	1 110 101 101		JSB	PRE11	GET RID OF INTEGER CIRCLE
127	0174	0313	1 100 110 001		JSB	ATC1	LOAD PI/4
128	0175		1 010 001 100		PT10		
129	0176	0233	1 001 110 001		JSB	PQO11	FORM PSEUDO QUOTIENT FOR TAN↑–1 1=PI/4
130	0177	0215	1 000 111 001		JSB	ATCD1	
131	0200		1 000 001 100		PT8		
132	0201	0234	1 001 110 101		JSB	PQO12	FORM PSEUDO QUOTIENT FOR TAN↑–1 10↑–1
133	0202		0 010 001 100		PT2		
134	0203		1 000 011 000		LDC8		
135	0204		0 110 001 100		PT6		
136	0205	0233	1 001 110 001		JSB	PQO11	FORM PSEUDO QUOTIENT FOR TAN↑–1 10↑–2
137	0206		0 100 001 100		PT4		
138	0207	0233	1 001 110 001		JSB	PQO11	FORM PSEUDO QUOTIENT FOR TAN↑–1 10↑–3
139	0210	0233	1 001 110 001		JSB	PQO11	FORM PSEUDO QUOTIENT FOR TAN↑–1 10↑–4
140	0211		1 100 101 110		W,AXB		
141	0212		1 001 001 110		W,SRC		
142	0213		1 101 001 100		PT13		
143	0214		0 101 011 000		LDC5		
144	0215	0373	1 111 101 111		BRN	TAN14	
145	0216		0 110 001 100	ATCD1	PT6		
146	0217		1 000 011 000		LDC8		
147	0220		0 110 011 000		LDC6		
148	0221		0 101 011 000		LDC5		
149	0222		0 010 011 000		LDC2		
150	0223		0 100 011 000		LDC4		
151	0224		1 001 011 000		LDC9		
152	0225		0 001 010 100	RTN11	YS1		
153	0226	0332	1 101 101 011		BRN	RTN12	
154	0227		0 000 110 000		RETURN		
155	0230		1 011 101 110	ADD10	W,ZTA		
156	0231		1 111 100 010		P,AP1A		
157	0232		0 010 010 000	ADD11	ROM 1		
158	0233		1 000 010 000	PMU11	ROM 4		
159	0234		0 100 001 110	PQO11	W,SLA		
160	0235		1 010 010 110	PQO12	MS,SRB		
161	0236		1 000 101 110		W,BXC		
162	0237	0241	1 010 000 111		BRN	PQO16	
163	0240		0 111 111 110	PQO15	S,CP1C		
164	0241		1 100 001 110	PQO16	W,AMBA		
165	0242	0240	1 010 000 011		BRN	PQO15	
166	0243		1 110 001 110		W,APBA		
167	0244		1 000 010 000	PQO13	ROM 4		
168	0245		1 000 010 000	MPY11	ROM 4		
169	0246		0 101 001 010	DIV11	X,AMCC		
170	0247		1 000 010 000		ROM 4		
171	0250		0 111 100 010	SQT15	P,CP1C		
172	0251		1 101 001 110	SQT16	W,AMCA		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0252	0250	1 010 100 011		BRN	SQT15
174	0253		1 111 001 110		W,APCA	
175	0254		0 100 001 110		W,SLA	
176	0255		0 000 011 100		PRS	
177	0256		1 001 010 010	SQT17	WP,SRC	
178	0257		0 000 101 100		YP0	
179	0260	0251	1 010 100 111		BRN	SQT16
180	0261	0055	0 010 110 111		BRN	TNM12
181	0262		0 111 100 010	DIV14	P,CP1C	
182	0263		1 100 010 110	DIV15	MS,AMBA	
183	0264	0262	1 011 001 011		BRN	DIV14
184	0265		1 110 010 110		MS,APBA	
185	0266		0 100 010 110		MS,SLA	
186	0267		0 000 011 100	DIV16	PRS	
187	0270		0 000 101 100		YP0	
188	0271	0263	1 011 001 111		BRN	DIV15
189	0272	0055	0 010 110 111		BRN	TNM12
190	0273		0 000 011 100	SQT12	PRS	
191	0274		1 110 010 110		MS,APBA	
192	0275	0333	1 101 101 111		BRN	SQT18
193	0276		1 110 010 000		ROM 7	
194	0277		0 101 111 010	ADD12	XS,CM1C	
195	0300		0 101 111 010		XS,CM1C	
196	0301		1 011 101 010		X,ZTA	
197	0302		1 101 011 110		S,AMCA	
198	0303		1 001 111 110		S,AM1	
199	0304	0306	1 100 011 011		BRN	ADD13
200	0305		1 000 010 000		ROM 4	
201	0306		1 000 000 110	ADD13	M,AMB	
202	0307	0312	1 100 101 011		BRN	ADD14
203	0310		0 011 111 110		S,ZNCC	
204	0311		1 100 101 110		W,AXB	
205	0312		1 100 001 110	ADD14	W,AMBA	
206	0313		1 000 010 000	ADD15	ROM 4	
207	0314		0 011 001 110	ATC1	W,ZTC	
208	0315		1 011 001 100		PT11	
209	0316		0 111 011 000		LDC7	
210	0317		1 000 011 000		LDC8	
211	0320		0 101 011 000		LDC5	
212	0321		0 011 011 000		LDC3	
213	0322		1 001 011 000		LDC9	
214	0323		1 000 011 000		LDC8	
215	0324		0 001 011 000		LDC1	
216	0325		0 110 011 000		LDC6	
217	0326		0 011 011 000		LDC3	
218	0327		0 101 011 000		LDC5	
219	0330		1 100 001 100		PT12	
220	0331		0 000 110 000		RETURN	
221	0332		0 100 010 000	RTN12	ROM 2	
222	0333		1 110 001 010	SQT18	X,APBA	
223	0334	0336	1 101 111 011		BRN	SQT14
224	0335		0 101 100 010		P,CM1C	
225	0336		0 111 111 110	SQT14	S,CP1C	
226	0337		0 000 101 100		YP0	
227	0340	0273	1 011 101 111		BRN	SQT12
228	0341		1 110 101 010		X,AXC	
229	0342		1 011 101 010		X,ZTA	
230	0343		0 001 100 010		P,CM1	
231	0344	0346	1 110 011 011		BRN	SQT13
232	0345		1 011 001 110		W,SRA	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		1 001 001 110	SQT13	W,SRC		
234	0347		1 000 101 010		X,BXC		
235	0350		0 011 001 010		X,ZTC		
236	0351		1 100 001 100		PT12		
237	0352	0256	1 010 111 011		BRN	SQT17	
238	0353		1 000 010 000	PRE11	ROM 4		CONTINUE IN ROM 4
239	0354		1 010 010 010	TAN18	WP,SRB		TAN14 TO TAN19 USES THE
240	0355		1 010 010 010		WP,SRB		PSEUDO QUOTIENTS FORMED
241	0356		0 101 111 110	TAN19	S,CM1C		BEFORE ON THE FOLLOWING
242	0357	0354	1 110 110 011		BRN	TAN18	
243	0360		0 111 010 010		WP,APCC		EQUATIONS
244	0361		1 100 010 010		WP,AMBA		
245	0362		1 000 110 010		WP,BXC		
246	0363		0 010 001 110	TAN13	W,BTC		
247	0364		1 101 111 110		S,AM1A		
248	0365	0356	1 110 111 011		BRN	TAN19	
249	0366		1 110 110 010		WP,AXC		
250	0367		0 110 101 000		STA		
251	0370		0 000 011 110		S,ZMB		
252	0371	0001	0 000 000 111		BRN	TAN15	
253	0372		0 100 001 110		W,SLA		
254	0373		1 110 110 010	TAN14	WP,AXC		
255	0374		0 100 101 000		CTS		
256	0375		1 010 010 010		WP,SRB		
257	0376		0 101 111 110		S,CM1C		
258	0377		1 000 111 110		S,BXC		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—4)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000		1 011 111 000		DSTC	RELOAD OLD M
4	0001	0374	1 111 110 011		BRN ERROR2	GO TO PRINT ERROR MESSAGE
5	0002		1 100 101 000	RTN41	DNR	
6	0003		0 000 110 000		RETURN	
7	0004		0 010 101 000	PROGM1	CMX	
8	0005		1 100 001 100		PT12	
9	0006		0 111 000 000	PROGM	PINC	
10	0007		1 000 100 000		READ	FROM ROM 0 PROGRAM MODE
11	0010		1 011 110 010		WP,ZTA	READ THE 2 TABLE DIGITS
12	0011		0 001 010 000		EERA	
13	0012		1 101 001 100	PROGM2	PT13	
14	0013		0 100 011 000	PROGM4	LDC4	
15	0014		1 101 001 100		PT13	
16	0015		0 001 000 010		P,AMC	
17	0016	0075	0 011 110 111		BRN CHECK	
18	0017	0300	1 100 000 011		BRN PROGM5	
19	0020		1 100 010 000	RTN610	ROM 6	
20	0021		1 100 101 000	RTN31	DNR	
21	0022		0 100 010 000		ROM 2	
22	0023		1 010 001 100	DIGT0	PT10	
23	0024		0 010 101 000		CXM	
24	0025		0 101 100 010		P,CMIC	
25	0026	0004	0 000 010 011		BRN PROGM1	
26	0027		0 010 101 000		CXM	
27	0030		1 000 101 110		W,BXC	
28	0031	0012	0 000 101 011		BRN PROGM2	
29	0032		1 111 100 010	DIGT3	P,APIA	
30	0033		1 111 100 010	DIGT2	P,APIA	
31	0034		1 111 100 010	DIGT1	P,APIA	
32	0035	0023	0 001 001 111		BRN DIGT0	
33	0036		0 010 010 000	PROGM3	ROM 1	GO TO ROUTINE BEGINNING
34	0037		1 010 010 000	RTN510	ROM 5	
35	0040		1 100 101 000	RTN51	DNR	
36	0041	0037	0 001 111 111		BRN RTN510	
37	0042		1 111 100 010	DIGT6	P,APIA	
38	0043		1 111 100 010	DIGT5	P,APIA	
39	0044		1 111 100 010	DIGT4	P,APIA	
40	0045	0032	0 001 101 011		BRN DIGT3	
41	0046		1 100 101 000	RTN61	DNR	
42	0047	0020	0 001 000 011		BRN RTN610	
43	0050		1 000 101 110	EEX	W,BXC	STORE C
44	0051	0056	0 010 111 011		BRN EEX0	
45	0052		1 111 100 010	DIGT9	P,APIA	
46	0053		1 111 100 010	DIGT8	P,APIA	
47	0054		1 111 100 010	DIGT7	P,APIA	
48	0055	0042	0 010 001 011		BRN DIGT6	
49	0056		1 010 101 000	EEX0	MTC	
50	0057		0 110 001 100		PT6	
51	0060		0 101 100 010		P,CMIC	CHECK IF COUNTER IS 0
52	0061	0063	0 011 001 111		BRN EEX01	NO, CONTINUE CHECK

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0316	1 100 111 011		BRN	EEX1	YES, INCREMENT COUNTER
54	0063		0 101 100 010	EEX01	P,CMIC		IS COUNTER 1
55	0064	0066	0 011 011 011		BRN	EEX02	NOM CONTINUE CHECK
56	0065	0316	1 100 111 011		BRN	EEX1	YES, INCREMENT COUNTER
57	0066		0 101 100 010	EEX02	P,CMIC		IS COUNTER 2
58	0067	0071	0 011 100 111		BRN	EEX03	NO, CONTINUE
59	0070	0316	1 100 111 011		BRN	EEX1	YES, GO TO INCREMENT COUNTER
60	0071		0 101 100 010	EEX03	P,CMIC		IS COUNTER 3
61	0072	0323	1 101 001 111		BRN	EEX3	NO, IT IS 4
62	0073		1 010 101 000		MTC		
63	0074	0350	1 110 100 011		BRN	EEX30	
64	0075		0 010 010 000	CHECK	ROM 1		PROGRAM MODE CHECK
65	0076		1 100 001 100	EEX5	PT12		
66	0077		1 001 010 010		WP,SRC		
67	0100	0333	1 101 110 001		JSB	EEX6	
68	0101		0 000 000 000		DUMMY		
69	0102	0170	0 111 100 011		BRN	NRM26	
70	0103		0 000 000 000		DUMMY		
71	0104		0 000 010 000	SUPCA	ROM 0		GO TO SUPERVISOR SKIP USERCODE CHECK
72	0105		0 000 010 000	SUPA4	ROM 0		
73	0106		0 000 000 000		DUMMY		
74	0107		0 001 100 110	DIV61	M,CM1		CHECK FOR ZERO X
75	0110	0116	0 100 111 011		BRN	DIV41	NO, GO TO DIVIDE
76	0111		1 100 001 100		PT12		
77	0112		0 001 011 000		LDC1		LOAD 1 IF X IS ZERO
78	0113		0 001 000 100		SS1		SET S1 TO INDICATE TRIG ROUTINE
79	0114		1 011 110 010		WP,ZTA		
80	0115		1 110 101 110		W,AXC		
81	0116		0 000 101 110	DIV41	W,ZTB		
82	0117		0 101 001 010		X,AMCC		
83	0120		1 100 001 100		PT12		
84	0121	0250	1 010 100 011		BRN	DIV21	
85	0122		1 100 101 000	SUPA40	DNR		
86	0123	0105	0 100 010 111		BRN	SUPA4	
87	0124	0367	1 111 011 111		BRN	EEXX1	EEX ROUTINE FROM ROM 1
88	0125		0 010 010 000	EEXX	ROM 1		CONTINUE EEX ROUTINE IN ROM 1
89	0126		0 010 101 000	SET	CXM		
90	0127		0 000 001 100		PT0		
91	0130		0 100 011 000		LDC4		
92	0131		0 010 101 000		CXM		
93	0132		0 000 110 000		RETURN		
94	0133		1 001 111 010	PRE29	XS,AM1		
95	0134	0302	1 100 001 011		BRN	PRE27	
96	0135		1 100 010 110	PRE24	MS,AMBA		
97	0136	0135	0 101 110 111		BRN	PRE24	
98	0137		1 110 010 110		MS,APBA		
99	0140		0 100 001 110		W,SLA		
100	0141		0 101 101 010		X,CMIC		
101	0142	0133	0 101 101 111		BRN	PRE29	
102	0143		1 011 001 110	PRE25	W,SRA		
103	0144		0 011 010 010		WP,ZTC		
104	0145		1 110 101 010		X,AXC		
105	0146		0 110 111 110	PRE26	S,ZMC		
106	0147	0153	0 110 101 111		BRN	PRE28	
107	0150		1 100 101 110		W,AXB		
108	0151		1 100 001 110		W,AMBA		
109	0152		0 011 101 010		X,ZNCC		
110	0153		1 011 001 110	PRE28	W,SRA		
111	0154		1 000 101 110	PQO23	W,BXC		
112	0155		0 011 001 110		W,ZTC		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 101 100 110		M,CMIC		
114	0157		0 000 000 000		NOP		
115	0160	0162	0 111 001 011		BRN	PQO28	
116	0161		0 110 011 000	PQO27	LDC6		
117	0162		0 001 101 100	PQO28	YP1		
118	0163	0161	0 111 000 111		BRN	PQO27	
119	0164		1 001 001 110		W,SRC		
120	0165		1 001 001 110	PQO24	W,SRC		
121	0166		0 000 000 000		NOP		
122	0167	0224	1 001 010 011		BRN	RTN21	
123	0170		0 000 001 100	NRM26	PT0		
124	0171		0 100 101 000		CTS		
125	0172		1 010 101 000		MTC		
126	0173		0 101 100 010		P,CMIC		
127	0174	0176	0 111 111 011		BRN	A1	
128	0175		0 000 000 000		DUMMY		
129	0176		0 101 100 010	A1	P,CMIC		
130	0177	0201	1 000 000 111		BRN	A2	
131	0200	0220	1 001 000 011		BRN	RTN11	RETURN IN ROM1
132	0201		0 101 100 010	A2	P,CMIC		
133	0202	0204	1 000 010 011		BRN	A3	
134	0203	0021	0 001 000 111		BRN	RTN31	RETURN IN ROM 2
135	0204		0 101 100 010	A3	P,CMIC		
136	0205	0207	1 000 011 111		BRN	A0	
137	0206	0222	1 001 001 011		BRN	RTN20	RETURN IN ROM 3
138	0207		0 101 100 010	A0	P,CMIC		
139	0210	0212	1 000 101 011		BRN	A4	
140	0211	0002	0 000 001 011		BRN	RTN41	RETURN IN ROM 4
141	0212		0 101 100 010	A4	P,CMIC		
142	0213	0215	1 000 110 111		BRN	A5	
143	0214	0040	0 010 000 011		BRN	RTN51	RETURN IN ROM 5
144	0215		0 101 100 010	A5	P,CMIC		
145	0216	0122	0 101 001 011		BRN	SUPA40	
146	0217	0046	0 010 011 011		BRN	RTN61	RETURN IN ROM 6
147	0220		1 100 101 000	RTN11	DNR		
148	0221		0 010 010 000		ROM 1		
149	0222		1 100 001 100	RTN20	PT12		
150	0223		1 100 101 000		DNR		
151	0224		0 110 010 000	RTN21	ROM 3		
152	0225		0 100 010 000	NRM21	ROM 2		
153	0226	0307	1 100 011 111		BRN	MPY28	
154	0227		0 000 000 000		DUMMY		
155	0230		0 000 000 000		DUMMY		
156	0231		1 100 001 100	ADD41	PT12		
157	0232		0 010 010 000		ROM 1		
158	0233		0 110 010 000	PQO21	ROM 3		
159	0234		1 011 001 110	PMU21	W,SRA		
160	0235		1 000 101 110	PMU22	W,BXC		
161	0236	0240	1 010 000 011		BRN	PMU24	
162	0237		1 110 001 110	PMU23	W,APBA		
163	0240		0 101 111 110	PMU24	S,CMIC		
164	0241	0237	1 001 111 111		BRN	PMU23	
165	0242		1 110 101 110		W,AXC		
166	0243		0 100 010 110		MS,SLA		
167	0244		1 110 101 110		W,AXC		
168	0245	0154	0 110 110 011		BRN	PQO23	
169	0246		0 011 001 100	MPY21	PT3		
170	0247		0 111 001 010	MPY22	X,APCC		
171	0250		0 101 011 110	DIV21	S,AMCC		
172	0251	0253	1 010 101 111		BRN	DIV22	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 010 111 110		S,ZMCC		
174	0253		1 100 100 110	DIV22	M,AXB		
175	0254		1 011 101 110		W,ZTA		
176	0255		1 100 101 100		YP12		
177	0256	0305	1 100 010 111		BRN	MPY27	
178	0257		0 001 100 110		M,CM1		
179	0260	0266	1 011 011 011		BRN	DEV23	
180	0261		0 001 010 100		YS1		
181	0262	0271	1 011 100 111		BRN	ERRØ	
182	0263		0 010 010 010		WP,BTC		
183	0264		1 101 100 110		M,AMIA		
184	0265		0 111 111 010		XS,CPIC		
185	0266		1 000 110 010	DIV23	WP,BXC		
186	0267		1 110 100 110		M,AXC		
187	0270		0 110 010 000		ROM 3		
188	0271		1 110 010 000	ERRØ	ROM 7		
189	0272		0 010 010 000	ERROR	ROM 1		
190	0273		0 010 010 000	EEX8	ROM 1		GO TO FINISH EEX ROUTINE
191	0274		0 011 011 110	EEX7	S,ZTC		
192	0275		0 010 101 010		X,ZMCC		COMPLIMENT EXP
193	0276	0272	1 011 101 101		JSB	EEX8	
194	0277	0271	1 011 100 111		BRN	ERRØ	
195	0300		0 011 100 100	PROGM5	RS3		
196	0301	0036	0 001 111 011		BRN	PROGM3	
197	0302		1 111 100 110	PRE27	M,APIA		
198	0303	0143	0 110 001 111		BRN	PRE25	
199	0304		1 110 001 110	MPY26	W,APBA		
200	0305		0 101 100 010	MPY27	P,CMIC		
201	0306	0304	1 100 010 011		BRN	MPY26	
202	0307		1 011 001 110	MPY28	W,SRA		
203	0310		0 000 111 100		PLS		
204	0311		1 101 101 100		YP13		
205	0312	0305	1 100 010 111		BRN	MPY27	
206	0313		0 111 101 010		X,CPIC		
207	0314		0 000 000 000		NOP		
208	0315	0225	1 001 010 111		BRN	NRM21	
209	0316		1 010 101 000	EEX1	MTC		INCREMENT COUNTER BY 1
210	0317		0 111 100 010		P,CPIC		
211	0320		0 010 101 000		CXM		
212	0321		1 000 101 110		W,BXC		RESTORE DATA
213	0322	0104	0 100 010 011		BRN	SUPCA	GO TO USPERVISOR SKIP USERCODE CHECK
214	0323		1 000 101 110	EEX3	W,BXC		RESTORE DATA
215	0324		0 001 111 010		XS,CM1		
216	0325	0272	1 011 101 011		BRN	ERROR	NEGATIVE,UNDERFLOW
217	0326		0 101 101 010		X,CMIC		
218	0327	0331	1 101 100 111		BRN	EEX4	
219	0330	0076	0 011 111 011		BRN	EEX5	1 DIGIT EXP
220	0331		0 101 101 010	EEX4	X,CMIC		
221	0332	0272	1 011 101 011		BRN	ERROR	
222	0333		1 100 001 100		PT12		
223	0334		1 001 010 010	EEX6	WP,SRC		
224	0335		0 000 011 100		PRS		
225	0336		0 001 101 100		YP1		
226	0337	0334	1 101 110 011		BRN	EEX6	
227	0340		0 001 111 110		SCM1		CHECK FOR SIGN
228	0341	0274	1 011 110 011		BRN	EEX7	NEGATIVE
229	0342	0273	1 011 101 111		BRN	EEX8	POSITIVE, GO TO FINISH ROUTINE
230	0343		1 011 111 000	EEX10	DSTC		LOAD ORIGINAL DATA
231	0344		1 110 101 010		X,AXC		LOAD NEW EXPONENT
232	0345		0 011 100 100		RS3		RESET OPERATION FLAG

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		1 100 001 100		PT12		RESET POINTER 12
234	0347	0105	0 100 010 111		BRN	SUPA4	
235	0350		0 111 100 010	EEX30	P,CPIC		INCREMENT COUNTER
236	0351		0 010 101 000		CXM		
237	0352		1 000 101 110		W,BXC		RESTORE DATA
238	0353		0 000 010 000	EQUAL	ROM 0		
239	0354		1 110 101 110	PRE21	W,AXC		
240	0355		0 100 101 110		W,ATB		
241	0356		0 110 000 110		M,CTA		
242	0357		1 010 111 010		XS,CPCC		
243	0360	0135	0 101 110 111		BRN	PRE24	
244	0361		0 111 111 010		XS,CPIC		
245	0362		1 011 001 110	PRE22	W,SRA		
246	0363		0 111 101 010		X,CPIC		
247	0364	0362	1 111 001 011		BRN	PRE22	
248	0365	0146	0 110 011 011		BRN	PRE26	
249	0366	0125	0 101 010 111		BRN	EEXX	FROM ROM 0
250	0367		1 011 110 000	EEXX1	DTDS		STORE M REGISTER IN DATA STORAGE
251	0370		1 001 001 100		PT9		
252	0371		0 000 011 000		LDC0		
253	0372		0 000 011 000		LDC0		LOAD 0 FOR = SIGN INTEPRETATION
254	0373		0 000 010 000		ROM 0		CONTINUE IN ROM 0
255	0374		0 010 101 000	ERROR2	CXM		RESTORE M
256	0374		0 011 001 110		W,ZTC		
257	0376		1 110 010 000		ROM 7		GO TO ROM 7 FOR ERROR MESSAGE 35
258	0377	0343	1 110 001 111		BRN	EEX10	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—5)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000		0 000 011 100	TEST1	PRS	TEST IF THERE ARE 2 DIGITS
4	0001		1 001 001 110		W,SRC	SHIFT RIGHT ONE IFF NOT
5	0002		1 010 101 100	TEST	YP10	
6	0003	0000	0 000 000 011		BRN TEST1	
7	0004		0 000 110 000		RETURN	
8	0005		0 011 001 110	ATC7	W,ZTC	
9	0006		1 100 001 100		PT12	
10	0007		0 001 011 000		LDC1	
11	0010		0 101 011 000		LDC5	LOAD 0.00015 FOR ROUND OFF
12	0011		0 011 011 000		LDC3	
13	0012		0 101 101 010		X,CM1C	
14	0013		0 101 101 010		X,CM1C	
15	0014		0 101 101 010		X,CM1C	
16	0015		0 101 101 010		X,CM1C	
17	0016		0 000 110 000		RETURN	
18	0017	0331	1 101 100 111		BRN DTRD2	
19	0020		0 100 010 000	ATCC0	ROM 2	LOAD PI
20	0021		0 000 001 100	SET	PT0	
21	0022		0 010 101 000		CXM	
22	0023		0 101 011 000		LDC5	
23	0024		0 010 101 000		CXM	
24	0025		0 000 110 000		RETURN	
25	0026		0 100 010 000	OFL1	ROM 2	
26	0027		0 011 001 110	ATC5	W,ZTC	LOAD 60
27	0030		1 100 001 100		PT12	
28	0031		0 110 011 000		LDC6	
29	0032		0 111 101 010		X,CP1C	
30	0033		1 100 001 100		PT12	
31	0034		0 000 110 000		RETURN	
32	0035		1 000 101 110	TEMP	W,BXC	RELOAD DATA
33	0036		0 110 101 000		STA	
34	0037		1 110 101 110		W,AXC	
35	0040		0 000 110 000		RETURN	
36	0041	0051	0 010 100 111		BRN TDWD	DMS TO DWDF
37	0042	0126	0 101 011 011		BRN TDMS	DWDF TO DMS
38	0043		1 000 000 100		SS8	GRAD TO DWDF
39	0044	0263	1 011 001 111		BRN DTGR	DWDF TO GRAD
40	0045		1 000 000 100		SS8	RADIAN TO DWDF
41	0046	0311	1 100 100 111		BRN DTRD	DWDF TO RADIAN
42	0047		1 000 000 100		SS8	MIL TO DWDF
43	0050	0340	1 110 000 011		BRN DTML	DWDF TO MIL
44	0051	0020	0 001 000 101	TDWD	JSB SET	SET RETURN FLAG
45	0052		0 010 001 100		PT2	
46	0053		1 011 101 110		W,ZTA	TAKE SECOND PART AND STORE THE REST
47	0054		1 110 110 010		WP,AXC	
48	0055		0 100 101 000		CTS	
49	0056	0110	0 100 100 101		JSB NORM1	SHIFT LEFT 10 PLACES
50	0057		1 111 101 010		X,AP1A	CORRECT EXPONENT
51	0060	0026	0 001 011 101		JSB ATC5	LOAD 60
52	0061	0247	1 010 100 001		JSB DIV51	DIVIDE

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0062	0034	0 001 110 101		JSB	TEMP	RELOAD DATA
54	0063		0 110 001 100		PT6		TAKE THE MINUTE PART
55	0064		1 011 101 110		W,ZTA		AND STORE THE REST
56	0065		1 110 110 010		WP,AXC		
57	0066		0 100 101 000		CTS		
58	0067		1 000 101 110		W,BXC		
59	0070	0110	0 100 100 101		JSB	NORM1	NORMALIZE IT
60	0071		1 111 101 010		X,AP1A		
61	0072	0230	1 001 100 101		JSB	ADD51	ADD DIVISION RESULT TO MINUTE
62	0073	0026	0 001 011 101		JSB	ATC5	LOAD 60
63	0074	0247	1 010 100 001		JSB	DIV51	DIVIDE MINUTE BY 60
64	0075	0034	0 001 110 101		JSB	TEMP	RELOAD DATA AND GET DEGREE PART
65	0076		0 110 001 110		W,CTA		
66	0077		0 110 101 110		W,ZMC		
67	0100	0106	0 100 011 011		BRN	TDWD1	
68	0101		1 100 001 100		PT12		
69	0102	0117	0 101 000 001		JSB	NORM4	CHECK FOR LEADING ZERO
70	0103		1 111 101 010		X,AP1A		CORRECT EXPONENT
71	0104		1 111 101 010		X,AP1A		ADD EXPONENT
72	0105		1 111 101 010		X,AP1A		
73	0106		1 000 101 110	TDWD1	W,BXC		
74	0107	0230	1 001 100 101		JSB	ADD51	ADD RESULT TO DEGREE
75	0110	0026	0 001 011 011		BRN	OFL1	GO TO OVERFLOW TEST AND DISPLAY
76	0111		0 100 101 110	NORM1	W,ATB		NORMALIZATION ROUTINE
77	0112		0 000 001 110		W,ZMB		CHECK IF DATA IS ZERO
78	0113	0125	0 101 010 111		BRN	NORM3	IF SO, RETURN
79	0114		0 100 001 110	NORM2	W,SLA		
80	0115		0 000 111 100		PLS		SHIFT LEFT UNTIL POINTER IS AT 12
81	0116		1 100 101 100		YP12		
82	0117	0114	0 100 110 011		BRN	NORM2	
83	0120		1 001 100 010	NORM4	P,AM1		CHECK FOR LEADING ZERO
84	0121	0125	0 101 010 111		BRN	NORM3	
85	0122		1 101 101 010		X,AM1A		
86	0123		0 100 010 110		MS,SLA		
87	0124	0120	0 101 000 011		BRN	NORM4	
88	0125		0 000 110 000	NORM3	RETURN		
89	0126	0020	0 001 000 101	TDMS	JSB	SET	SET RETURN FLAG
90	0127		0 110 001 110		W,CTA		
91	0130		0 001 111 110		S,CM1		
92	0131	0240	1 010 000 011		BRN	TDMS02	
93	0132	0004	0 000 010 101	TDMS01	JSB	ATC7	LOAD 0.000153 FOR ROUND OFF
94	0133	0230	1 001 100 101		JSB	ADD51	ADD ROUND OFF NUMBER
95	0134		0 001 111 010		XS,CM1		CHECK IF DATA IS LESS THAN 1
96	0135	0225	1 001 010 111		BRN	TDMS1	YES,CONVERT TO MINUTE
97	0136		0 011 001 110		W,ZTC		
98	0137		0 000 001 100		PT0		CHECK IF NUMBER IS TOO LARGE
99	0140		0 100 011 000		LDC4		
100	0141		0 001 001 010		X,AMC		
101	0142	0376	1 111 111 011		BRN	ERR1	IF SO GO TO ERROR MESSAGE
102	0143		1 100 101 110		W,AXB		IF NOT, CONTINUE
103	0144		0 010 001 110		W,BTC		
104	0145		1 100 001 100		PT12		
105	0146	0252	1 010 101 101		JSB	XRM1	DETERMINE SIZE OF DEGREE PART
106	0147	0152	0 110 101 011		BRN	TDMS7	
107	0150		0 000 011 100	TDMS8	PRS		IF LESS THAN 4 DIGITS MOVE
108	0151		1 001 001 110		W,SRC		TO RIGHT ACCORDINGLY
109	0152		1 000 101 100	TDMS7	YP8		
110	0153	0150	0 110 100 011		BRN	TDMS8	
111	0154		0 100 101 000		CTS		
112	0155		0 010 001 110		W,BTC		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156		1 100 001 100		PT12			
114	0157	0252	1 010 101 101		JSB	XRM1		RESET POINTER TO DECIMAL LOCATION
115	0160	0110	0 100 100 101		JSB	NORM1		NORMALIZE DECIMAL PART
116	0161		1 101 101 010		X,AM1A			
117	0162	0026	0 001 001 101	TDMS6	JSB	ATC5		LOAD 60
118	0163	0243	1 010 010 001		JSB	MPY51		MULTIPLY
119	0164		0 001 111 010		XS,CM1			CHECK IF MINUTE PART IS LESS THAN 1
120	0165	0205	1 000 010 111		BRN	TDMS2		YES, CONVERT TO SECOND
121	0166		0 100 101 110		W,ATB			STORE DATA IN B
122	0167		1 100 001 100		PT12			
123	0170	0252	1 010 101 101		JSB	XRM1		SAVE INTEGER PART AS MINUTE
124	0171	0001	0 000 001 001		JSB	TEST		TEST IF MINUTE HAS 2 DIGITS
125	0172		1 100 001 100		PT12			
126	0173		1 001 001 110	TDMS3	W,SRC			
127	0174		0 000 011 100		PRS			
128	0175		0 110 101 100		YP6			MOVE MINUTE PART TO CORRECT LOCATION
129	0176	0173	0 111 101 111		BRN	TDMS3		
130	0177	0232	1 001 101 101		JSB	STORE		SAVE MINUTE PART IN PROPER LOCATION
131	0200		0 010 001 110		W,BTC			RESTORE DATA IN C
132	0201		1 100 001 100		PT12			
133	0202	0252	1 010 101 101		JSB	XRM1		SET POINTER TO DECIMAL LOCATION
134	0203	0110	0 100 100 101		JSB	NORM1		NORMALIZE DECIMAL PART
135	0204		1 101 101 010		X,AM1A			CORRECT EXPONENT
136	0205	0026	0 001 011 101	TDMS2	JSB	ATC5		LOAD 60
137	0206	0243	1 010 010 001		JSB	MPY51		
138	0207		0 001 111 010		XS,CM1			CHECK IF SECOND PART IS LESS THAN 1
139	0210	0222	1 001 001 011		BRN	TDMS5		IF SO, RESULT IS ZERO,DONE.
140	0211		1 100 001 100		PT12			
141	0212	0252	1 010 101 101		JSB	XRM1		DETERMINE SIZE OF SECOND PATT
142	0213	0001	0 000 001 001		JSB	TEST		TEST IF IT HAS 2 DIGITS
143	0214		1 100 001 100		PT12			
144	0215		1 001 001 110	TDMS4	W,SRC			MOVE SECOND PART INTO RIGHT LOCATION
145	0216		0 000 011 100		PRS			
146	0217		0 010 101 100		YP2			
147	0220	0215	1 000 110 111		BRN	TDMS4		
148	0221	0232	1 001 101 101		JSB	STORE		STORE RESULT
149	0222		0 110 101 000	TDMS5	STA			
150	0223	0310	1 100 100 011		BRN	PRINT		GO TO PRINT ANGLE ROUTINE
151	0224		0 000 000 000		DUMMY			
152	0225		0 110 001 110	TDMS1	W,CTA			STORE ZERO INTO DEGREE
153	0226		0 011 001 110		W,ZTC			PART AND CONVERT TO SECOND
154	0227		0 100 101 000		CTS			
155	0230	0162	0 111 001 011		BRN	TDMS6		
156	0231		1 100 001 100	ADD51	PT12			
157	0232		0 010 010 000		ROM 1			ADD ROUTINE
158	0233		0 110 101 000	STORE	STA			STORE PROPERLY LOCATED DATA IN STACK
159	0234		0 110 010 010		WP,CTA			
160	0235		1 110 101 110		W,AXC			
161	0236		0 100 101 000		CTS			
162	0237		0 000 110 000		RETURN			
163	0240	0365	1 111 011 001	TDMS02	JSB	ATC6		
164	0241		1 110 101 110		W,AXC			
165	0242	0230	1 001 100 101		JSB	ADD51		
166	0243	0132	0 101 101 011		BRN	TDMS01		
167	0244		0 000 101 110	MPY51	W,ZTB			
168	0245		1 000 010 000		ROM 4			
169	0246		0 101 001 010	DIV510	X,AMCC			
170	0247		1 000 010 000		ROM 4			
171	0250		0 000 101 110	DIV51	W,ZTB			
172	0251	0246	1 010 011 011		BRN	DIV510		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
173	0252		0 000 011 100	XRM2	PRS			
174	0253		0 101 101 010	XRM1	X,CM1C		ISOLATE INTEGER PART	
175	0254	0252	1 010 101 011		BRN	XRM2		
176	0255		0 000 011 100		PRS			
177	0256		0 011 001 010		X,ZTC			
178	0257		1 011 101 110		W,ZTA			
179	0260		0 110 010 010		WP,CTA			
180	0261		0 011 010 010		WP,ZTC			
181	0262		0 000 110 000		RETURN			
182	0263	0020	0 001 000 101	DTGR	JSB	SET	SET RETURN FLAG	
183	0264		1 110 101 110		W,AXC		DEGREE TO GRAD OR GRAD TO DEGREE CONVERSION	
184	0265	0365	1 111 011 001		JSB	ATC6	LOAD 360	
185	0266		1 000 010 100		YS8			
186	0267	0272	1 011 101 011		BRN	DIVV	IF SET CONVERT GRAD TO DWDF	
187	0270	0243	1 010 010 001		JSB	MPY51		
188	0271	0273	1 011 101 111		BRN	CONT1		
189	0272	0247	1 010 100 001	DIVV	JSB	DIV51		
190	0273		1 011 101 110	CONT1	W,ZTA			
191	0274		1 110 101 110		W,AXC			
192	0275		1 100 001 100		PT12			
193	0276		0 100 011 000		LDC4		LOAD 400	
194	0277		0 000 001 100		PT0			
195	0300		0 010 011 000		LDC2			
196	0301		1 100 001 100		PT12			
197	0302		1 000 010 100		YS8		IF S8 NOT SET CONVERT DWDF TO GRAD	
198	0303	0306	1 100 011 011		BRN	MPYY	IF SET CONVERT GRAD TO DWDF	
199	0304	0247	1 010 100 001		JSB	DIV51		
200	0305	0307	1 100 011 111		BRN	CONT2		
201	0306	0243	1 010 010 001	MPYY	JSB	MPY51		
202	0307	0026	0 001 011 011	CONT2	BRN	OFL1		
203	0310		1 110 010 000	PRINT	ROM 7			
204	0311	0020	0 001 000 101	DTRD	JSB	SET	SET RETURN FLAG	
205	0312		1 011 101 110		W,ZTA		DEGREE TO RADIAN OR RADIAN TO DEGREE	
206	0313		1 110 101 110		W,AXC			
207	0314		1 100 001 100		PT12			
208	0315		0 001 011 000		LDC1			
209	0316		1 000 011 000		LDC8			
210	0317		0 000 001 100		PT0			
211	0320		0 010 011 000		LDC2			
212	0321		1 100 001 100		PT12			
213	0322		1 000 010 100		YS8		IF S8 NOT SET CONVERT RAD TO DWDF	
214	0323	0326	1 101 011 011		BRN	DIVV1	IF SET CONVERT DWDF TO RAD	
215	0324	0243	1 010 010 001		JSB	MPY51		
216	0325	0327	1 101 011 111		BRN	DTRD1		
217	0326	0247	1 010 100 001	DIVV1	JSB	DIV51		
218	0327		0 011 000 100	DTRD1	SS3		SET LOAD PI FLAG	
219	0330	0020	0 001 000 011		BRN	ATCC0	GO TO LOAD PI	
220	0331		0 011 100 100	DTRD2	RS3		RESET LOAD PI FLAG	
221	0332		1 000 010 100		YS8		IF S8 NOT SET CONVERT DWDF TO RAD	
222	0333	0336	1 101 111 011		BRN	MPYY1	IF SET CONVERT RAD TO DWDF	
223	0334	0247	1 010 100 001		JSB	DIV51		
224	0335	0337	1 101 111 111		BRN	DTRD3		
225	0336	0243	1 010 010 001	MPYY1	JSB	MRY51		
226	0337	0026	0 001 011 011	DTRD3	BRN	OFL1	DONE	
227	0340	0020	0 001 000 101	DTML	JSB	SET	SET RETURN FLAG	
228	0341		1 110 101 110		W,AXC		DWDF TO MIL OR MIL TO DWDF	
229	0342	0365	1 111 011 001		JSB	ATC6	LOAD 360	
230	0343		1 000 010 100		YS8		IF S8 NOT SET CONVERT DWDF TO MIL	
231	0344	0347	1 110 011 111		BRN	DIVV2	IF SET CONVERT MIL TO DWDF	
232	0345	0243	1 010 010 001		JSB	MPY51		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346	0350	1 110 100 011		BRN	DTML1	
234	0347	0247	1 010 100 001	DIVV2	JSB	DIV51	
235	0350		1 011 101 110	DTML1	W,ZTA		
236	0351		1 110 101 110		W,AXC		
237	0352		1 100 001 100		PT12		
238	0353		0 110 011 000		LDC6		
239	0354		0 100 011 000		LDC4		LOAD 6400
240	0355		0 000 001 100		PT0		
241	0356		0 100 011 000		LDC4		
242	0357		1 100 001 100		PT12		
243	0360		1 000 010 100		YS8		
244	0361	0364	1 111 010 011		BRN	MPYY2	
245	0362	0247	1 010 100 001		JSB	DIV51	
246	0363	0365	1 111 010 111		BRN	DTML2	
247	0364	0243	1 010 010 001	MPYY2	JSB	MPY51	
248	0365	0026	0 001 011 011	DTML2	BRN	OFL1	DONE
249	0366		0 011 001 110	ATC6	W,ZTC		LOAD 360
250	0367		1 100 001 100		PT12		
251	0370		0 011 011 000		LDC3		LOAD 360
252	0371		0 110 011 000		LDC6		
253	0372		0 000 001 100		PT0		
254	0373		0 010 011 000		LDC2		
255	0374		1 100 001 100		PT12		
256	0375		0 000 110 000		RETURN		
257	0376		1 110 010 000	ERR1	ROM 7		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—6)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000		0 010 010 000	SIN	ROM 1	
4	0001		0 010 010 000	COS	ROM 1	
5	0002		0 011 111 110	ACC30	S,ZNCC	
6	0003		0 000 000 000		NOP	
7	0004	0367	1 111 011 111		BRN	ACC4
8	0005		0 010 010 000	ATAN	ROM 1	
9	0006		0 010 010 000	SQRT	ROM 1	
10	0007		0 000 000 000		DUMMY	
11	0010		1 000 101 110	ADDR	W,BXC	SAVE C
12	0011		0 011 001 110		W,ZTC	PREPARE TO GENERATE DS ADDRESS
13	0012		0 111 101 110		W,CP1C	
14	0013		1 100 001 100		PT12	
15	0014		1 010 011 000		LDC10	LOAD DS ADDRESS
16	0015		0 000 110 000		RETURN	
17	0016		0 000 011 000	ADDR0	LDC0	LOAD DS REGISTER ADDRESS
18	0017		1 001 110 000	ADDRC	ATDS	LOAD DS ADDRESS
19	0020		0 010 001 110		W,BTC	RESTORE C
20	0021		0 000 110 000		RETURN	
21	0022		0 001 011 000	ADDR1	LDC1	LOAD DS REGISTER ADDRESS
22	0023	0017	0 000 111 111		BRN	ADDRC CONTINUE
23	0024		0 000 010 000	PRINT	ROM 0	
24	0025		0 000 000 000		DUMMY	
25	0026	0063	0 011 001 111		BRN	XEXY X EXCHANGE WITH Y ROUTINE
26	0027		0 100 000 100	PRXY	SS4	PRINT X,Y ROUTINE
27	0030	0260	1 011 000 011		BRN	POPUT
28	0031	0321	1 101 000 111		BRN	RECAL RECALL ACCUMULATED VALUE ROUTINE
29	0032	0220	1 001 000 011		BRN	CLEAR CLEAR ACCUMULATED VALUE ROUTINE
30	0033		0 010 001 110	TRECT	W,BTC	TO RECTANGULAR ROUTINE
31	0034		0 100 101 000		CTS	SAVE ANGLE
32	0035		0 010 000 100		SS2	SET RETURN FROM TRIG FLAG
33	0036	0001	0 000 000 111		BRN	COS COMPUTE COS @
34	0037	0007	0 000 100 001	RCOS	JSB	ADDR
35	0040	0015	0 000 111 001		JSB	ADDR0 LOAD DS ADDRESS
36	0041		1 011 111 000		DSTC	LOAD R IN C
37	0042		0 100 101 000		CTS	SAVE R IN STACK
38	0043	0243	1 010 010 001		JSB	MPY61 COMPUTE X BY R COS @
39	0044		1 011 110 000		DTDS	STORE X IN DS REGISTER 0
40	0045		1 100 101 000		DNR	LOAD R IN C
41	0046		0 110 101 000		STA	LOAD @ IN A
42	0047		0 100 101 000		CTS	
43	0050		1 110 101 110		W,AXC	LOAD @ IN C
44	0051		0 010 000 100		SS2	
45	0052	000	0 000 000 011		BRN	SIN COMPUTE SIN @
46	0053		0 110 101 000	RSIN	STA	LOAD R TO A
47	0054	0243	1 010 010 001		JSB	MPY61 COMPUTE Y BY R SIN @
48	0055	0007	0 000 100 001		JSB	ADDR
49	0056	0021	0 001 001 001		JSB	ADDR1 LOAD DS DS REGISTER 1 ADDRESS
50	0057		1 011 110 000		DTDS	STORE Y IN DS 1
51	0060		0 100 000 100		SS4	SET FLAG FOR SYMBOL X,Y
52	0061	0257	1 010 111 111		BRN	OPUT GO TO OUPUT

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0246	1 010 011 011		BRN	PRRA	
54	0063	0007	0 000 100 001	XEXY	JSB	ADDR	
55	0064	0021	0 001 001 001		JSB	ADDR1	LOAD DS REGISTER 1 ADDRESS
56	0065		1 011 111 000		DSTC		LOAD Y TO C
57	0066		0 100 101 000		CTS		SAVE Y
58	0067	0007	0 000 100 001		JSB	ADDR	
59	0070	0015	0 000 111 001		JSB	ADDR0	LOAD DS REGISTER 0 ADDRESS
60	0071		1 011 111 000		DSTC		LOAD X TO C
61	0072		0 110 101 000		STA		LOAD Y IN A
62	0073		1 110 101 110		W,AXC		Y IN C AND X IN A
63	0074		1 011 110 000		DTDS		LOAD Y IN DS REGISTER 0
64	0075		1 110 101 110		W,AXC		Y IN A AND X IN C
65	0076	0007	0 000 100 001		JSB	ADDR	
66	0077	0021	0 001 001 001		JSB	ADDR1	LOAD DS REGISTER 1 ADDRESS
67	0100		1 011 110 000		DTDS		LOAD X IN DS 1
68	0101	0105	0 100 010 111		BRN	SUPA6	
69	0102		0 010 101 000	SUPA60	CXM		
70	0103		0 000 110 100	SUPA61	CLS		
71	0104		0 111 000 100		SS7		
72	0105		0 000 010 000	SUPA6	ROM 0		
73	0106		1 000 010 000	DIV61	ROM 4		
74	0107		0 010 001 110	TPOLR	W,BTC		TO POLAR ROUTINE
75	0110		0 100 101 000		CTS		SAVE Y
76	0111		0 110 001 110		W,CTA		
77	0112	0174	0 111 110 101		JSB	SET	SET RETURN FLAG
78	0113	0243	1 010 010 001		JSB	MPY61	COMPUTE Y2
79	0114		0 100 101 000		CTS		SAVE Y2
80	0115	0007	0 000 100 001		JSB	ADDR	
81	0116	0015	0 000 111 001		JSB	ADDR0	LOAD DS REGISTER 0 ADDRESS
82	0117		1 011 111 000		DSTC		LOAD X IN C
83	0120		0 110 001 110		W,CTA		
84	0121	0243	1 010 010 001		JSB	MPY61	COMPUTE X2
85	0122		0 110 101 000		STA		LOAD Y2 IN A
86	0123	0230	1 001 100 101		JSB	ADD61	ADD THE TWO
87	0124		0 010 000 100		SS2		
88	0125	0006	0 000 011 011		BRN	SQRT	COMPUTE R
89	0126		1 011 111 000	RSQRT	DSTC		LOAD X IN C
90	0127		1 110 101 110		W,AXC		LOAD X IN C
91	0130		1 011 110 000		DTDS		STORE R IN DS REGISTER 0
92	0131		1 110 101 110		W,AXC		LOAD R IN C, S IN A
93	0132		0 110 101 000		STA		Y IN A
94	0133		0 001 111 110		S,CM1		CHECK IF X IS NEGATIVE
95	0134	0137	0 101 111 111		BRN	TPOLR1	YES, PREPARE TO ADD 180
96	0135		1 000 100 100		RS8		NO, RESET 8
97	0136	0140	0 110 000 011		BRN	TPOLR2	
98	0137		1 000 000 100	TPOLR1	SS8		
99	0140	0105	0 100 011 001	TPOLR2	JSB	DIV61	FIND Y/X
100	0141		0 010 000 100		SS2		
101	0142	0005	0 000 010 111		BRN	ATAN	COMPUTE ARCTAN Y/X
102	0143		1 000 010 100	RATAN	YS8		IS S8 SET
103	0144	0147	0 110 011 111		BRN	TPOLR3	NO, CONTINUE
104	0145	0164	0 111 010 101		JSB	ATC1	YES, LOAD 180
105	0146	0230	1 001 100 101		JSB	ADD61	ADD180
106	0147		0 001 111 110	TPOLR3	S,CM1		CHECK IF ANGLE IS NEGATIVE
107	0150	0152	0 110 101 011		BRN	TPOLR4	YES, ADD 360
108	0151	0155	0 110 110 111		BRN	TPOLR5	NO, CONTINUE
109	0152	0164	0 111 010 101	TPOLR4	JSB	ATC1	LOAD 180
110	0153		1 010 100 110		M,CPCC		ADD TO 360
111	0154	0230	1 001 100 101		JSB	ADD61	ADD 360 TO RESULT
112	0155	0007	0 000 100 001	TPOLR5	JSB	ADDR	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156	0021	0 001 001 001		JSB	ADDR1	LOAD DS REGISTER 1 ADDRESS
114	0157		1 011 110 000		DTDS		STORE ANGLE IN DS REGISTER 1
115	0160		0 100 100 100		RS4		SET FLAG TO LOAD SYMBOL A AND R
116	0161	0257	1 010 111 111		BRN	OPUT	GO TO OUTPUT
117	0162		0 010 011 000	ADDR2	LDC2		LOAD DS REGISTER ADDESS 2
118	0163	0017	0 000 111 111		BRN	ADDRC	
119	0164	0027	0 001 011 111		BRN	PRXY	PRINT X,Y GROM ROM 1
120	0165		0 011 001 110	ATC1	W,ZTC		
121	0166		1 100 001 100		PT12		
122	0167		0 001 011 000		LDC1		LOAD 180
123	0170		1 000 011 000		LDC8		
124	0171		0 000 001 100		PT0		
125	0172		0 010 011 000		LDC2		
126	0173		0 000 110 000		RETURN		
127	0174	0275	1 011 110 111		BRN	OPUT3	PRINT DATA RETURN
128	0175		0 010 101 000	SET	CXM		
129	0176		0 000 001 100		PT0		
130	0177		0 110 011 000		LDC6		SET RETURN FLAG
131	0200		0 010 101 000	SET1	CXM		
132	0201		0 000 110 000		RETURN		
133	0202	0107	0 100 011 111		BRN	TPOLR	TO POLAR ROUTINE FROM ROM 1
134	0203	0033	0 001 101 111		BRN	TRECT	TO RECT. ROUTINE FROM ROM 1
135	0204	0337	1 101 111 111		BRN	ACC+	ACC+ ROUTINE FROM ROM 1
136	0205	0340	1 110 000 011		BRN	ACC–	ACC– ROUTINE FROM ROM 1
137	0206		0 011 011 000	ADDR3	LDC3		
138	0207	0017	0 000 111 111		BRN	ADDRC	LOAD DS REGISTER ADDRESS 3
139	0210		1 111 011 000	SYMBL	LDC15		
140	0211		0 000 101 100		YP0		
141	0212	0210	1 000 100 011		BRN	SYMBL	
142	0213	0240	1 010 000 011		BRN	SYMBL0	
143	0214	0053	0 010 101 111		BRN	RSIN	RETURN AFTER SIN
144	0215	0037	0 001 111 111		BRN	RCOS	RETURN AFTER COS
145	0216	0143	0 110 001 111		BRN	RATAN	RETURN AFTER ACTAN
146	0217	0126	0 101 011 011		BRN	RSQRT	RETURN AFTER SQUARE ROOT
147	0220		1 000 101 110	CLEAR	W,BXC		SAVE B
148	0221		0 011 001 110		W,ZTC		LOAD 0 IN C
149	0222	0007	0 000 100 001		JSB	ADDR	
150	0223	0161	0 111 001 001		JSB	ADDR2	LOAD DS REGISTER 2 ADDRESS
151	0224		1 011 110 000		DTDS		LOAD 0 IN DS REGISTER 2
152	0225	0007	0 000 100 001		JSB	ADDR	
153	0226	0205	1 000 011 001		JSB	ADDR3	LOAD DS REGISTER 3 ADDRESS
154	0227		1 011 110 000		DTDS		LOAD 0 IN DS 3
155	0230	0321	1 101 000 111		BRN	RECAL	OUTPUT THE 0 CONTENT
156	0231		1 100 001 100	ADD61	PT12		
157	0232		0 010 010 000		ROM 1		
158	0233		0 010 101 000	CHECK	CXM		CHECK FOR PROGRAM MODE ROUTINE
159	0234		1 100 001 100		PT12		
160	0235		0 001 100 010		P,CM1		
161	0236	0102	0 100 001 011		BRN	SUPA60	
162	0237	0253	1 010 101 111		BRN	CHECK1	CONTINUE TO CHECK PRINT ON COMMAND FLAG
163	0240		1 110 010 000	SYMBL0	ROM 7		PRINT SYMBOL ROUTINE
164	0241		0 000 000 000		DUMMY		
165	0242		1 011 100 100	CHECK2	RS11		
166	0243	0102	0 100 001 011		BRN	SUPA60	GO BACK TO SUPERVISOR
167	0244		0 000 101 110	MPY61	W,ZTB		
168	0245		1 000 010 000		ROM 4		
169	0246		0 100 100 100	PRRA	RS4		
170	0247	0260	1 011 000 011		BRN	POPUT	PRINT DS 0,1 WITH SYMBOL R,A
272	0250		1 110 101 110	ACC3	W,AXC		COMPLIMENT ENTERED X
172	0251	0002	0 000 001 011		BRN	ACC30	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0252		0 000 000 000		DUMMY	
174	0253		1 110 100 000	CHECK1	YPOC	CHECK PRINT ON COMMAND FLAG
175	0254		1 011 010 100		YS11	
176	0255	0200	1 000 000 011		BRN	SET1 NOT SET, CONTINUE WITH OUTPUT
177	0256	0242	1 010 001 011		BRN	CHECK2 SET, GO TO SUPERVISOR
178	0257	0232	1 001 101 101	OPUT	JSB	CHECK CHECK FOR PROGRAM MODE
179	0260		0 010 101 000	POPUT	CXM	
180	0261		0 001 001 100		PT1	
181	0262		0 001 011 000		LDC1	
182	0263		0 010 101 000		CXM	
183	0264		0 100 010 100		YS4	IS S4 SET
184	0265	0311	1 100 100 111		BRN	POLAR NO, PREPARE SYMBOL R
185	0266		1 101 001 100		PT13	
186	0267		0 000 011 000		LDC0	YES, LOAD SYMBOL X
187	0270	0210	1 000 100 011	SYMBL1	BRN	SYMBL GO TO ROM 7 TO PRINT SYMBOL
188	0271	0007	0 000 100 001	OPUT1	JSB	ADDR
189	0272	0015	0 000 111 001		JSB	ADDR0 LOAD DS REGSTER 0 ADDRESS
190	0273		1 011 111 000	OPUT4	DSTC	LOAD X OR R IN C
191	0274	0024	0 001 010 011		BRN	PRINT GO TO ROM A DATA PRINT ROUTINE
192	0275		0 001 001 100	OPUT3	PT1	
193	0276		0 010 101 000		CXM	LOAD SECOND PRINT RETURN FLAG
194	0277		0 010 011 000		LDC2	
195	0300		0 010 101 000		CXM	
196	0301		0 100 010 100		YS4	S4 SET
197	0302	0314	1 100 110 011		BRN	POLAR1 NO, GO TO LOAD SYMBOL A
198	0303		1 101 001 100		PT13	YES, LOAD SYMBOL Y
199	0304		0 001 011 000		LDC1	
200	0305	0210	1 000 100 011	SYMBL2	BRN	SYMBL GO TO ROM 7 AND PRINT SYMBOL
201	0306	0007	0 000 100 001	OPUT2	JSB	ADDR
202	0307	0021	0 001 001 001		JSB	ADDR1 LOAD DS REGISTER 1 ADDRESS
203	0310	0273	1 011 101 111		BRN	OPUT4 LOAD Y OR A IN C
204	0311		1 101 001 100	POLAR	PT13	LOAD SYMBOL R
205	0312		1 011 011 000		LDC11	
206	0313	0270	1 011 100 011		BRN	SYMBL1 GO TO PRINT SYJBOL AND DATA
207	0314		1 101 001 100	PLAR1	PT13	LOAD SYMBOL A
208	0315		0 100 011 000		LDC4	
209	0316	0305	1 100 010 111		BRN	SYMBL2 GO TO PRINT SYMBOL AND DATA
210	0317	0271	1 011 100 111		BRN	OPUT1 RETURN FROM PRINT SYMBOL IN ROM 7
211	0320	0306	1 100 011 011		BRN	OPUT2 RETURN FROM PRINT SYMBOL IN ROM 7
212	0321	0007	0 000 100 001	RECAL	JSB	ADDR RECALL ACC ROUTINE
213	0322	0161	0 111 001 001		JSB	ADDR2 LOAD DS REGISTER 2 ADDRESS
214	0323		1 011 111 000		DSTC	LOAD X IN C
215	0324	0007	0 000 100 001		JSB	ADDR
216	0325	0015	0 000 111 001		JSB	ADDR0 LOAD DS REGISTER 0 ADDRESS
217	0326		1 011 110 000		DTDS	LOAD X IN DS REGISTER 0
218	0327	0007	0 000 100 001		JSB	ADDR
219	0330	0205	1 000 011 001		JSB	ADDR3 LOAD DS REGISTER 3 ADDRESS
220	0331		1 011 111 000		DSTC	LOAD Y IN C
221	0332	0007	0 000 100 001		JSB	ADDR
222	0333	0021	0 001 001 001		JSB	ADDR1 LOAD DS REGISTER 1 ADDRESS
223	0334		1 011 110 000		DTDS	LOAD Y IN DS REGISTER 1
224	0335		0 100 000 100		SS4	
225	0336	0257	1 010 111 111		BRN	OPUT GO TO OUTPUT
226	0337		1 000 000 100	ACC+	SS8	
227	0340		0 010 001 110	ACC–	W,BTC	
228	0341		0 100 101 000		CTS	SAVE ENTERED Y IN STACK
229	0342	0174	0 111 110 101		JSB	SET RETURN FLAG
230	0343	0007	0 000 100 001		JSB	ADDR
231	0344	0021	0 001 001 001		JSB	ADDR1 LOAD DS REGISTER 1 ADDRESS
232	0345		1 011 110 000		DTDS	STORE ENTERED IN DS REGISTER 1

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 110 001 110		W,CTA		
234	0347	0007	0 000 100 001		JSB	ADDR	
235	0350	0205	1 000 011 001		JSB	ADDR3	LOAD DS REGISTER 3 ADDRESS
236	0351		1 011 111 000		DSTC		LOAD Y IN C
237	0352		1 000 010 100		YS8		IS S8 SET
238	0353	0374	1 111 110 011		BRN	ACC1	NO, IT IS ACC–
239	0354	0230	1 001 100 101	ACC2	JSB	ADD61	YES, GO TO ADD
240	0355		1 011 110 000		DTDS		STORE FINAL Y IN DS REGISTR 3
241	0356	0007	0 000 100 001		JSB	ADDR	
242	0357	0015	0 000 111 001		JSB	ADDR0	LOAD DS EGISTER 0 ADDRESS
243	0360		1 011 111 000		DSTC		LOAD X IN C?
244	0361		0 110 001 110		W,CTA		LOAD X TO A
245	0362	0007	0 000 100 001		JSB	ADDR	
246	0363	0161	0 111 001 001		JSB	ADDR2	LOAD DS REGISTER 2 ADDRESS
247	0364		1 011 111 000		DSTC		
248	0365		1 000 010 100		YS8		IS S8 SET
249	0366	0250	1 010 100 011		BRN	ACC3	NO, IT IS ACC–, COMPLIMENT
250	0367	0230	1 001 100 101	ACC4	JSB	ADD61	YES, GO TO ADD
251	0370		1 011 110 000		DTDS		STORE FINAL X IN DS REGISTER 2
252	0371		1 000 100 100		RS8		RESET S8
253	0372		1 100 101 000		DNR		RESTORE ENTERED Y
254	0373	0105	0 100 010 111		BRN	SUPA6	GO TO SUPERVISOR
255	0374		1 110 101 110	ACC1	W,AXC		
256	0375		0 011 111 110		S,ZNCC		COMPLIMENT ENTERED Y
257	0376		0 000 000 000		NOP		
258	0377	0354	1 110 110 011		BRN	ACC2	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—7)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 000 001 100	FPRT	PT0		
4	0001		0 101 110 000		PRE		PRINTER ENABLE
5	0002		1 011 010 100	MESB	YS11		FLAG?
6	0003	0002	0 000 001 011		BRN	MESB	NO,WAIT
7	0004		1 011 100 100		RS11		YES,RESET IT
8	0005		1 111 110 000		CCS		RIGHT PART OF PRINTER MASK
9	0006		1 101 110 000		TCS		LEFT PART OF PRINTER MASK
10	0007		0 000 111 100		PLS		INCREMENT SECTOR COUNTER
11	0010		1 101 101 100	MESBB	YP13		LAST SECTOR
12	0011	0002	0 000 001 011		BRN	MESB	NO
13	0012		1 011 010 100	MESC	YS11		YES,FLAG
14	0013	0012	0 000 101 011		BRN	MESC	NO,WAIT
15	0014		1 011 100 100		RS11		YES,RESET
16	0015		0 011 110 000		ADV		PAPER ADVANCE
17	0016		0 000 110 000		RETURN		
18	0017		1 101 001 100	NOTE	PT13		GENERATE MESSAGE "NOTE"
19	0020		1 000 101 110		W,BXC		SAVE C REG.
20	0021		1 111 011 000	MESA	LDC15		PREPARE "NOTE 19" MESSAGE
21	0022		0 111 101 100		YP7		
22	0023	0021	0 001 000 111		BRN	MESA	
23	0024		1 100 011 000		LDC12		N
24	0025		1 111 011 000		LDC15		
25	0026		0 000 011 000		LDC0		O
26	0027		1 111 011 000		LDC15		
27	0030		1 011 011 000		LDC11		T
28	0031		1 111 011 000		LDC15		
29	0032		1 011 011 000		LDC11		E
30	0033		1 111 011 000		LDC15		
31	0034		0 111 110 000		CTT		
32	0035		0 000 110 000		RETURN		
33	0036		1 101 001 100	ENTER	PT13		
34	0037		1 111 011 000		LDC15		LOAD SPACE
35	0040		0 000 000 000		NOP		
36	0041		1 011 011 000		LDC11		E
37	0042		1 100 011 000		LDC12		N
38	0043		1 010 011 000		LDC10		T
39	0044		1 011 011 000		LDC11		E
40	0045		1 011 011 000		LDC11		R
41	0046		1 111 011 000		LDC15		
42	0047		0 000 110 000	HOME	RETURN		
43	0050		1 111 011 000	PRANG7	LDC15		
44	0051		0 001 001 100	PRANG3	PT1		
45	0052		1 111 011 000		LDC15		
46	0053		1 111 011 000		LDC15		SECOND
47	0054		0 111 110 000		CTT		LEFT PART OF PRINT MASK
48	0055		0 010 001 100		PT2		
49	0056		0 100 001 110	PRANG4	W,SLA		
50	0057		0 000 111 100		PLS		
51	0060		1 101 101 100		YP13		
52	0061	0056	0 010 111 011		BRN	PRANG4	

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
53	0062		1 110 101 110		W,AXC	LOAD IT IN C
54	0063		0 001 100 010		P,CM1	CHECK FOR LEADING ZERO
55	0064	0066	0 011 011 011		BRN PRANG5	NO, CONTINUE
56	0065		1 111 011 000		LDC15	YES, BLANK IT
57	0066		1 011 001 100	PRANG5	PT11	
58	0067		1 111 011 000		LDC15	
59	0070		1 111 011 000		LDC15	
60	0071	7777	0 000 000 001		JSB FPRT	GO TO PRINT IT
61	0072		1 100 101 110		W,AXB	RELOAD DATA
62	0073		1 010 101 000		MTC	
63	0074		1 100 001 100		PT12	
64	0075		0 001 100 010		P,CM1	
65	0076	0113	0 100 101 111		BRN SUPA70	
66	0077		0 100 000 100		SS4	
67	0100	0351	1 110 100 111		BRN DISPLY	
68	0101		0 000 010 000	PRINTA	ROM 0	
69	0102	0332	1 101 101 011		BRN ERR5	PROGRAM MODE TABLE NUMBER ERROR
70	0103	0016	0 000 111 101		JSB NOTE	OVERFLOW ERROR MESSAGE
71	0104	0106	0 100 011 011		BRN ERR3	
72	0105		0 000 010 000	SUPA7	ROM 0	
73	0106		1 101 001 100	ERR3	PT13	
74	0107		0 011 011 000		LDC3	
75	0110		0 000 000 000		NOP	
76	0111		0 001 011 000		LDC1	
77	0112	0125	0 101 010 111		BRN ERR2	
78	0113		1 110 101 110	SUPA70	W,AXC	
79	0114		0 000 010 100	WAIT	YS0	
80	0115	0362	1 111 001 011		BRN OPUT9	FINISH ROUTINE CLEAR STATUS ETC
81	0116		0 000 100 100		RS0	WAIT FOR KEY UP
82	0117	0114	0 100 110 011		BRN WAIT	
83	0120	0016	0 000 111 101		JSB NOTE	ERROR NOT 19 ROUTINE FROM ROM 0
84	0121		1 101 001 100		PT13	
85	0122		0 011 011 000		LDC13	
86	0123		0 000 000 000		NOP	
87	0124		0 000 011 000		LDC0	
88	0125	7777	0 000 000 001	ERR2	JSB FPRT	
89	0126		0 010 001 110		W,BTC	
90	0127		0 010 101 000		CXM	
91	0130		1 100 001 100		PT12	
92	0131		0 000 011 000		LDC0	
93	0132		0 010 101 000		CXM	
94	0133	0362	1 111 001 011		BRN OPUT9	TO RESET STATUS BIT
95	0134		0 000 000 000		DUMMY	
96	0135		0 000 000 000		DUMMY	
97	0136	0317	1 101 000 001	OPUT	JSB CHECK1	
98	0137		0 100 101 000		CTS	
99	0140		1 101 001 100		PT13	
100	0141		1 111 011 000	OPUT1	LDC15	
101	0142		0 000 101 100		YP0	
102	0143	0141	0 110 000 111		BRN OPUT1	
103	0144		1 111 011 000		LDC15	
104	0145	0244	1 010 010 011		BRN OPUT3	
105	0146		1 111 011 000	PRANG6	LDC15	
106	0147		0 001 100 010		P,CM1	
107	0150	0155	0 110 110 111		BRN PRANG1	NO, CONTINUE
108	0151		1 111 011 000		LDC15	YES, BLANK IT
109	0152		0 001 100 010		P,CM1	CHECK FOR LEADING ZERO
110	0153	0155	0 110 110 111		BRN PRANG1	NO, CONTINUE
111	0154		1 111 011 000		LDC15	YES, BLANK IT
112	0155		0 101 001 100	PRANG1	PT5	BLANK BETWEEN DEGREE AND MINUTE

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		1 111 011 000		LDC15		
114	0157		1 111 011 000		LDC15		
115	0160		0 001 100 010		P,CM1		CHECK FOR LEADING ZERO ON MINUTE
116	0161	0051	0 010 100 111		BRN	PRANG3	NO, CONTINUE
117	0162	0050	0 010 100 011		BRN	PRANG7	
118	0163	0351	1 110 100 111		BRN	DISPLY	GO TO DISPLY GROM ROM 0
119	0164		0 110 001 110	PRANGG	W,CTA		
120	0165	0345	1 110 010 111		BRN	PRAN	GO TO CHECK PRINT ON COMMAND FLAG
121	0166		1 001 001 110	PRANN	W,SRC		ALIGNMENT SHIFT
122	0167		1 001 001 110		W,SRC		
123	0170		1 001 001 110		W,SRC		
124	0171		1 101 001 100		PT13		
125	0172		1 111 011 000	PRANG0	LDC15		LOAD LEADING SPACES
126	0173		1 001 101 100		YP9		
127	0174	0172	0 111 101 011		BRN	PRANG0	
128	0175		0 001 100 010		P,CM1		CHECK FOR LEADING ZERO ON DEGREE
129	0176	0155	0 110 110 111		BRN	PRANG1	NOM CONTINUE
130	0177	0146	0 110 011 011		BRN	PRANG6	
131	0200	0242	1 010 001 011		BRN	PRIN0	
132	0201		1 111 011 000	PRINT	LDC15		
133	0202		0 011 101 100		YP3		
134	0203	0201	1 000 000 111		BRN	PRINT	LOAD SPACE FOR PRINTER
135	0204		1 011 011 000		LDC11		T
136	0205		1 011 011 000		LDC11		A
137	0206		0 000 011 000		LDC0		B
138	0207		1 111 011 000		LDC15		
139	0210		0 111 110 000		CTT		
140	0211		1 110 101 110		W,AXC		
141	0212		0 110 001 110		W,CTA		
142	0213		1 011 001 100		PT11		
143	0214		1 111 011 000		LDC15		
144	0215		1 111 011 000		LDC15		
145	0216	7777	0 000 000 001		JSB	FPRT	
146	0217	0344	1 110 010 011		BRN	TAB	GO BACK TO TABLE ROUTINE IN ROM 1
147	0220		0 000 000 000		DUMMY		
148	0221	0035	0 001 111 001	PRINT1	JSB	ENTER	LOAD MESSAGE "ENTER"
149	0222		1 111 011 000	PRINT3	LDC15		
150	0223		0 010 101 100		YP2		
151	0224	0222	1 001 001 011		BRN	PRINT3	
152	0225		1 011 011 000		LDC11		A
153	0226		1 100 011 000		LDC12		N
154	0227		1 000 011 000		LDC8		G
155	0230		0 111 110 000	PRINT4	CTT		
156	0231		1 101 001 100		PT13		
157	0232		1 111 011 000	PRINT5	LDC15		
158	0233		1 001 101 100		YP9		
159	0234	0232	1 001 101 011		BRN	PRINT5	
160	0235	7777	0 000 000 001		JSB	FPRT	
161	0236		1 011 101 110	PRINT6	W,ZTA		CLEAR A FOR ZERO DISPLAY
162	0237		0 000 000 000		NOP		TO PREVENT CAARRY
163	0240	0351	1 110 100 111		BRN	DISPLY	
164	0241	0256	1 010 111 011		BRN	SYMBL	FROM ROM 6
165	0242	0324	1 101 010 101	PRIN0	JSB	CHECK2	
166	0243	0221	1 001 000 111		BRN	PRINT1	GO TO PRINT ANGLE
167	0244		0 111 110 000	OPUT3	CTT		
168	0245		0 110 101 000		STA		
169	0246	0305	1 100 010 111		BRN	OPUT5	
170	0247	0136	0 101 111 011		BRN	OPUT	
171	0250	0016	0 000 111 101	ERR1	JSB	NOTE	ZERO DIVISOR ERROR ROUTINE
172	0251		1 101 001 100		PT13		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 011 011 000		LDC3		
174	0253		0 000 000 000		NOP		
175	0254		0 010 011 000		LDC2		
176	0255	0125	0 101 010 111		BRN	ERR2	
177	0256		1 111 011 000	SYMBL	LDC15		
178	0257		0 111 110 000		CTT		
179	0260		1 101 001 100		PT13		
180	0261		1 111 011 000	SYMBL2	LDC15		
181	0262		1 001 101 100		YP9		
182	0263	0261	1 011 000 111		BRN	SYMBL2	
183	0264	7777	0 000 000 001		JSB	FPRT	
184	0265		1 010 101 000		MTC		
185	0266		0 001 001 100		PT1		
186	0267	0273	1 011 101 111		BRN	SYMBL3	
187	0270		1 011 100 100	CHECK4	RS11		
188	0271	0351	1 110 100 111		BRN	DISPLY	PRINT ON COMMAND SET GO TO DISPLAY WITHOUT PRINT
189	0272	0250	1 010 100 011		BRN	ERR1	
190	0273		0 101 100 010	SYMBL3	P,CM1C		
191	0274		0 101 100 010		P,CM1C		
192	0275	0317	1 100 111 111		BRN	SYMBL4	
193	0276	0316	1 100 111 011		BRN	SYMBL5	
194	0277	0016	0 000 111 101		JSB	NOTE	SQUARROOT OF A NEGATIVE NUMBER
195	0300		1 101 001 100		PT13		
196	0301		0 011 011 000		LDC3		LOAD NOTE 30
197	0302		0 000 000 000		NOP		
198	0303		0 011 011 000		LDC3		
199	0304	0125	0 101 010 111		BRN	ERR2	
200	0305		0 100 001 110	OPUT5	W,SLA		
201	0306		1 110 101 110		W,AXC		
202	0307		1 101 001 100		PT13		
203	0310	0313	1 100 101 111		BRN	OPUT6	
204	0311		1 110 101 110		W,AXC		
205	0312	0164	0 111 010 011		BRN	PRANGG	
206	0313		1 100 011 000	OPUT6	LDC12		
207	0314		1 011 001 100		PT11		
208	0315	0354	1 110 110 011		BRN	OPUT7	
209	0316		1 100 010 000	SYMBL5	ROM 6		RETURN TO ROM 6 AFTER SYMBOL PRINT 1
210	0317		1 100 010 000	SYMBL4	ROM 6		RETURN TO ROM 6 AFTER SYMBOL PRINT 2
211	0320		1 110 100 000	CHECK1	YPOC		PRINT ON COMMAND FLAG CHECK
212	0321		1 011 010 100		YS11		
213	0322	0047	0 010 011 111		BRN	HOME	NOT SET, GO BACK TO PRINT STORE AND RECAL SYMBOL
214	0323		1 011 100 100		RS11		
215	0324	0365	1 111 010 111		BRN	SUPA71	
216	0325		1 110 100 000	CHECK2	YPOC		PRINT ON COMMAND FLAG CHECK
217	0326		1 011 010 100		YS11		
218	0327	0047	0 010 011 111		BRN	HOME	NOTSET, GO BACK TO PRINT ENTER ANGLE
219	0330		1 011 100 100		RS11		
220	0331	0236	1 001 111 011		BRN	PRINT6	SET, GO TO ANGLE DISPLAY WITHOUT PRINT
221	0332	0016	0 000 111 101	ERR5	JSB	NOTE	TABLE NUMBER ERROR
222	0333		1 101 001 100		PT13		
223	0334		0 011 011 000		LDC3		
224	0335		0 000 000 000		NOP		
225	0336		0 100 011 000		LDC4		
226	0337	0125	0 101 010 111		BRN	ERR2	
227	0340		1 110 100 000	CHECK3	YPOC		PRINT ON COMMAND CHECK
228	0341		1 011 010 100		YS11		
229	0342	0047	0 010 011 111		BRN	HOME	NOT SET, GO BACK TO ANGLE PRINT
230	0343	0270	1 011 100 011		BRN	CHECK4	SET, GO TO DISPLY
231	0344		0 010 010 000	TAB	ROM 1		
232	0345		0 100 101 110	PRAN	W,ATB		

MATH/USER-DEFINABLE FUNCTION BLOCK LISTING (ROM B—7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346	0337	1 110 000 001		JSB	CHECK3	CHECK PRINT ON COMMAND FLAG
234	0347	0166	0 111 011 011		BRN	PRANN	NOT SET, GO BACK TO PRINT ANGLE
235	0350		0 000 000 000		DUMMY		
236	0351		0 100 010 000	DISPLY	ROM 2		
237	0352		0 010 001 110	CLEAR3	W,BTC		
238	0353	0105	0 100 010 111		BRN	SUPA7	
239	0354		1 111 011 000	OPUT7	LDC15		
240	0355		0 110 010 100		YS6		
241	0356	0367	1 111 011 111		BRN	OPUT2	
242	0357		0 011 011 000		LDC3		
243	0360	7777	0 000 000 001	OPUT8	JSB	FPRT	
244	0361		0 010 001 110		W,BTC		
245	0362		0 000 110 100	OPUT9	CLS		
246	0363		0 111 000 100		SS7		SET FIRST DIGIT ENTRY FLAG
247	0364	0105	0 100 010 111		BRN	SUPA7	
248	0365		1 000 101 110	SUPA71	W,BXC		
249	0366	0105	0 100 010 111		BRN	SUPA7	
250	0367		0 100 011 000	OPUT2	LDC4		
251	0370	0360	1 111 000 011		BRN	OPUT8	
252	0371	0016	0 000 111 101	ERR4	JSB	NOTE	NUMBER TOO LARGE ERROR FOR CONVERSION
253	0372		1 101 001 100		PT13		
254	0373		0 011 011 000		LDC3		
255	0374		0 000 000 000		NOP		
256	0375		0 101 011 000		LDC5		
257	0376	0125	0 101 010 111		BRN	ERR2	
258	0377	0371	1 111 100 111		BRN	ERR4	

STATISTICS FUNCTION BLOCK LISTING (ROM 0)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000	0175	0 111 110 111		BRN	SKIP—	0
4	0001	0351	1 110 100 111		BRN	TESPRT	
5	0002	0146	0 110 011 011		BRN	XROOT	
6	0003		0 000 100 000	PON	RMGRA		
7	0004	0003	0 000 001 111		BRN	PON	POWER ON ACCESS TO RMGRB
8	0005	0257	1 011 000 001	PLOT	JSB	PREP	
9	0006		1 000 001 100		PT8		
10	0007		1 010 010 000		ROM 5		
11	0010	0257	1 011 000 001	KEY1	JSB	PREP	10 HIST KEY ENTRY
12	0011		1 010 010 000		ROM 5		
13	0012	0005	0 000 010 111	KEY2	BRN	PLOT	PLOT KEY ENTRY
14	0013	0043	0 010 001 111	KEY3	BRN	CT	CHAR# KEY ENTRY
15	0014	0257	1 011 000 001	KEY4	JSB	PREP	AXES KEY ENTRY
16	0015		1 100 010 000		ROM 6		
17	0016	0257	1 011 000 001	KEY5	JSB	PREP	OFFSET/WIDTH KEY ENTRY
18	0017	0323	1 101 001 111		BRN	OFST	
19	0020	0257	1 011 000 001	KEY6	JSB	PREP	20 LINEAR KEY ENTRY
20	0021		0 110 010 000		ROM 3		
21	0022		1 100 010 000	KEY7	ROM 6		RECALL KEY ENTRY
22	0023	0072	0 011 101 011	KEY8	BRN	T	T KEY ENTRY
23	0024	0257	1 011 000 001	KEY9	JSB	PREP	SAMPLE KEY ENTRY
24	0025		1 100 010 000		ROM 6		
25	0026	0257	1 011 000 001	KEY10	JSB	PREP	VAR KEY ENTRY
26	0027		1 000 010 000		ROM 4		
27	0030	0257	1 011 000 001	KEY11	JSB	PREP	30 PAR KEY ENTRY
28	0031		0 110 010 000		ROM 3		
29	0032	0144	0 110 010 011	KEY12	BRN	EVAL	EVAL KEY ENTRY
30	0033	0237	1 001 111 111	KEY13	BRN	BASIC	BASIC KEY ENTRY
31	0034	0257	1 011 000 001	KEY14	JSB	PREP	DELETE KEY ENTRY
32	0035		1 100 010 000		ROM 6		
33	0036	0257	1 011 000 001	KEY15	JSB	PREP	DATA ENTRY KEY ENTRY
34	0037		1 100 010 000		ROM 6		
35	0040		1 010 010 100	FBKD	YS10		40
36	0041	0373	1 111 101 111		BRN	AK	
37	0042	0207	1 000 011 111		BRN	PRGMR	
38	0043	0257	1 011 000 001	CT	JSB	PREP	CHAR# KEY
39	0044		0 010 101 000		CXM		
40	0045		1 011 001 100		PT11		
41	0046		0 101 011 000		LDC5		
42	0047	0345	1 110 010 111		BRN	EC	
43	0050		0 000 110 100	RETROUT	CLS		50 RETURN FROM EXT. ROUTINE
44	0051		0 100 101 000		CTS		DATA TO D
45	0052		0 111 001 100		PT7		
46	0053	0361	1 111 001 001		JSB	CH1A	53 STAT FLAG TO M
47	0054		0 010 101 000		CXM		HENRY'S FLAG TO R17
48	0055		1 011 110 000		DTDS		
49	0056		1 100 101 000		DNR		DATA TO C
50	0057		0 101 000 000	CONT	IS2		
51	0060		0 111 000 000		PINC		60 INCREMENT FROM 2ND BIT
52	0061		0 111 000 000		PINC		OF PRGM COUNTER

STATISTICS FUNCTION BLOCK LISTING (ROM 0) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0062		0 101 101 000		SRI		NEW INSTRUCTION ADDRESS TO LS BITS OF IO	
54	0063		1 001 000 000		IS1			
55	0064		0 010 101 000		CXM		FLAGS TO A	
56	0065		0 110 001 110		W,CTA			
57	0066		0 010 101 000		CXM			
58	0067		0 000 000 000		DUMMY			
59	0070		0 111 001 100		PT7		70	
60	0071		0 010 010 000		ROM1		SEARCH ROM 1 FOR PRGM	
61	0072	0257	1 011 000 001	T	JSB	PREP		
62	0073	0173	0 111 101 111		BRN	T4		
63	0074	0347	1 110 100 001	XPRINT	JSB	EXM712		
64	0075		0 000 100 000		RMGRA			
65	0076		0 100 001 100	SUITE3	PT4			
66	0077		1 001 100 010		P,AM1		WHICH SAMPLE IS BEING ENTERED?	
67	0100	0103	0 100 001 111		BRN	DN2	Y	
68	0101		0 000 001 100	EX30	PT0		X	
69	0102	0104	0 100 010 011		BRN	DN1		
70	0103		1 001 001 100	DN2	PT9		FETCH N2	
71	0104	0360	1 111 000 101	DN1	JSB	CH0A		
72	0105		0 111 000 100	OUT	SS7		SET FIRST DIGIT FLAG	
73	0106		0 000 100 000		RMGRA		RETURN TO HENRY'S SUPERVISOR	
74	0107		1 011 111 110	NOTE	S,ZTA			
75	0110	0112	0 100 101 101		JSB	TO	110	
76	0111		1 111 111 110	INC	S,AP1A		SET AB TO PREVIOUSLY SET POINTER VALUE	
77	0112		0 000 011 100		PRS			
78	0113		0 000 101 100	TO	YP0			
79	0114	0111	0 100 100 111		BRN	INC		
80	0115	0366	1 111 011 101	NOTE30	JSB	BLANK	FILL C WITH BLANKS	
81	0116		1 101 001 100		PT13		3 TO C13	
82	0117		0 011 011 000		LDC3			
83	0120	0371	1 111 101 001		JSB	ADV	120 ADVANCE PAPER	
84	0121		1 011 001 110		W,SRA		SHIFT DIGIT IN A13 TO A12	
85	0122		1 110 100 010		P,AXC		A12 TO C12	
86	0123		0 010 101 000		CXM			
87	0124		0 000 011 000		LDC0		0 TO F12 and F11	
88	0125		0 000 011 000		LDC0			
89	0126		0 010 101 000		CXM			
90	0127	0366	1 111 011 101		JSB	BLANK	127 C TO T AND BLANK C	
91	0130		0 111 001 100		PT7		130 LOAD "NOTE" IN C	
92	0131		1 100 011 000		LDC12			
93	0132		0 101 001 100		PT5			
94	0133		0 000 011 000		LDC0			
95	0134		0 011 001 100		PT3			
96	0135		1 011 011 000		LDC11			
97	0136		0 001 001 100		PT1			
98	0137		1 011 011 000		LDC11			
99	0140	0370	1 111 100 101		JSB	PRINT	140 PRINT "NOTE 3X"	
100	0141		0 111 010 100		YS7		IS THIS NOTE 30?	
101	0142	0217	1 000 111 111		BRN	FIN	NO, GO THROUGH NORMAL EXZT	
102	0143	0101	0 100 000 111		BRN	EX30	YES, STRAIGHT OUT	
103	0144	0257	1 011 000 001	EVAL	JSB	PREP		
104	0145		1 010 010 000		ROM 5			
105	0146	0347	1 110 100 001	XROOT	JSB	EXM712	EXCHANGE FLAGS	
106	0147		1 100 001 100		PT12		1/2 TO C	
107	0150		0 011 001 110		W,ZTC		150	
108	0151		0 101 011 000		LDC5			
109	0152		0 011 001 100		PT3			
110	0153		0 101 101 010		X,CM1C			
111	0154		1 000 101 110		W,BXC		1/2 TO B	
112	0155		1 010 101 000		MTC		HENRY'S FLAG TO C	

(144)

STATISTICS FUNCTION BLOCK LISTING (ROM Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	Ø156		Ø ØØØ Ø11 ØØØ		LDCØ		Ø--> F3
114	Ø157		Ø 1Ø1 Ø11 ØØØ		LDC5		5--> F2
115	Ø16Ø		Ø ØØØ 1ØØ ØØØ		RMGRA		16Ø EXIT TO X TO Y
116	Ø161		Ø Ø1Ø 1Ø1 ØØØ	START	CXM		STAT FLAGS TO M
117	Ø162		Ø 111 11Ø ØØØ		CTT		LOAD STARTING ADDRESS TO PRGM COUNTER
118	Ø163		Ø 1Ø1 ØØØ ØØØ		IS2		
119	Ø164		Ø 1ØØ ØØ1 ØØØ		SLT		
12Ø	Ø165		1 ØØØ ØØØ ØØØ		TTP		
121	Ø166		1 ØØ1 ØØØ ØØØ		IS1		
122	Ø167		1 Ø1Ø 1ØØ ØØØ		SBL		SET BUSY LIGHT
123	Ø17Ø		1 ØØØ 1Ø1 ØØØ		DSOF		TURN DISPLAY OFF
124	Ø171		Ø ØØØ 1Ø1 ØØØ		DSTO		TURN DISPLAY ON
125	Ø172	Ø057	Ø Ø1Ø 111 111		BRN	CONT	
126	Ø173		1 ØØØ Ø1Ø ØØØ	T4	ROM 4		
127	Ø174		Ø ØØØ ØØØ ØØØ		DUMMY		
128	Ø175		Ø 1Ø1 ØØØ ØØØ	SKIP–	IS2		DECREMENT THE PRGM COUNTER FROM
129	Ø176		1 ØØ1 ØØ1 ØØØ		PDEC		THE 2ND BIT A NUMBER OF TIMES
13Ø	Ø177		1 ØØ1 ØØ1 ØØØ		PDEC		EQUAL TO THE PREVIOUSLY SET POINTER
131	Ø2ØØ		1 ØØ1 ØØØ ØØØ		IS1		2ØØ
132	Ø2Ø1		Ø ØØØ Ø11 1ØØ		PRS		
133	Ø2Ø2		Ø ØØØ 1Ø1 1ØØ		YPØ		
134	Ø2Ø3	Ø175	Ø 111 11Ø 111		BRN	SKIP–	2Ø3
135	Ø2Ø4	Ø057	Ø Ø1Ø 111 111		BRN	CONT	
136	Ø2Ø5	Ø1Ø7	Ø 1ØØ Ø11 111		BRN	NOTE	RETURN FROM ROM 7 ROUTINES
137	Ø2Ø6		Ø ØØØ 11Ø ØØØ		RETURN		
138	Ø2Ø7		1 Ø11 111 11Ø	PRGMR	S,ZTA		Ø TO A13 FOR NOTE 3Ø GENERATION
139	Ø21Ø		Ø 1Ø1 ØØØ ØØØ		IS2		21Ø
14Ø	Ø211	Ø177	1 ØØØ ØØØ ØØ1		LIO	4ØØ	
141	Ø212		Ø 11Ø 1ØØ ØØØ		GIOE		EGRESS TO PRGMR IN IO3
142	Ø213		1 ØØ1 ØØØ ØØØ		IS1		PRGMR NOT CONNECTED PRINT NOTE 3Ø
143	Ø214		Ø ØØØ 11Ø 1ØØ		CLS		SET FLAG FOR NOTE 3Ø TERMINATION
144	Ø215		Ø 111 ØØØ 1ØØ		SS7		
145	Ø216	Ø115	Ø 1ØØ 11Ø 111		BRN	NOTE3Ø	
146	Ø217		1 ØØØ ØØ1 1ØØ	FIN	PT8		FETCH LAST PRGM ADDRESS
147	Ø22Ø	Ø361	1 111 ØØ1 ØØ1		JSB	CH1A	22Ø AND RESTORE IT TO
148	Ø221		Ø 111 11Ø ØØØ		CTT		THE PRGM COUNTER.
149	Ø222		1 ØØØ ØØØ ØØØ		TTP		
15Ø	Ø223		Ø 111 ØØ1 1ØØ		PT7		HENRY'S FLAGS TO C
151	Ø224	Ø361	1 111 ØØ1 ØØ1		JSB	CH1A	
152	Ø225		Ø Ø1Ø 1Ø1 ØØØ		CXM		HENRY'S FLAGS TO M; STAT FLAGS TO C
153	Ø226		1 Ø11 11Ø ØØØ		DTDS		STORE STAT FLAGS IN R17
154	Ø227		1 11Ø 1Ø1 11Ø		W,AXC		STAT FLAGS TO A
155	Ø23Ø		Ø Ø1Ø 1Ø1 ØØØ		CXM		23Ø HENRY'S FLAGS TO C
156	Ø231		1 11Ø 1ØØ Ø1Ø		P,AXC		
157	Ø232		Ø 111 ØØ1 1ØØ		PT7		
158	Ø233		Ø Ø11 ØØØ Ø1Ø		P,ZTC		SET FLAG FOR MAIN UNIT CONTROL
159	Ø234		1 Ø11 ØØ1 1ØØ		PT11		IMPOSE DIGIT INTERP. ON HENRY
16Ø	Ø235		1 11Ø 1ØØ Ø1Ø		P,AXC		
161	Ø236	Ø355	1 11Ø 11Ø 111		BRN	SUITE1	
162	Ø237	Ø257	1 Ø11 ØØØ ØØ1	BASIC	JSB	PREP	
163	Ø24Ø		Ø 1ØØ Ø1Ø ØØØ		ROM 2		24Ø
164	Ø241	ØØ1Ø	Ø ØØØ 1ØØ Ø11		BRN	KEY1	ACCESS TO STAT FUNCTIONS
165	Ø242	ØØ12	Ø ØØØ 1Ø1 Ø11		BRN	KEY2	IN PROGRAM MODE
166	Ø243	ØØ13	Ø ØØØ 1Ø1 111		BRN	KEY3	
167	Ø244	ØØ14	Ø ØØØ 11Ø Ø11		BRN	KEY4	
168	Ø245	ØØ16	Ø ØØØ 111 Ø11		BRN	KEY5	
169	Ø246	ØØ2Ø	Ø ØØ1 ØØØ Ø11		BRN	KEY6	
17Ø	Ø247	ØØ22	Ø ØØ1 ØØ1 Ø11		BRN	KEY7	
171	Ø25Ø	ØØ23	Ø ØØ1 ØØ1 111		BRN	KEY8	25Ø
172	Ø251	ØØ24	Ø ØØ1 Ø1Ø Ø11		BRN	KEY9	

STATISTICS FUNCTION BLOCK LISTING (ROM 0) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252	0026	0 001 011 011		BRN	KEY10	
174	0253	0030	0 001 100 011		BRN	KEY11	
175	0254	0032	0 001 101 011		BRN	KEY12	
176	0255	0033	0 001 101 111		BRN	KEY13	
177	0256	0034	0 001 110 011		BRN	KEY14	
178	0257	0036	0 001 111 011		BRN	KEY15	257
179	0260		1 100 101 110	PREP	W,AXB		260 DIGIT ENTRY TO B
180	0261		0 110 101 000		STA		D TO A
181	0262		0 100 101 000		CTS		DATA TO D
182	0263		0 011 001 110		W,ZTC		ADDRESS R19
183	0264		0 111 101 110		W,CP1C		
184	0265		1 100 001 100		PT12		
185	0266		1 011 011 000		LDC11		
186	0267		0 000 110 100		CLS		
187	0270		0 101 100 010		P,CM1C		270
188	0271		1 001 110 000		ATDS		
189	0272		0 100 101 000		CTS		ADDRESS TO D
190	0273		1 110 101 110		W,AXC		
191	0274		1 011 110 000		DTDS		STORE D-REG IN R19
192	0275		1 100 101 000		DNR		BRING DOWN ADDRESS
193	0276		0 101 100 010		P,CM1C		DECREMENT ADDRESS
194	0277		1 001 110 000		ATDS		ADDRESS R18
195	0300		0 100 101 000		CTS		300 ADDRESS TO D
196	0301		0 101 000 000		IS2		PRGM ADDRESS TO C
197	0302		1 000 101 000		PTT		
198	0303		1 001 000 000		IS1		
199	0304		0 111 111 000		TTC		
200	0305		1 011 110 000		DTDS		STORE PRGM ADDRESS IN R18
201	0306		1 100 101 000		DNR		BRING DOWN ADDRESS
202	0307		0 101 100 010		P,CM1C		DECREMENT ADDRESS
203	0310		1 001 110 000		ATDS		310 ADDRESS R17
204	0311		1 100 001 100		PT12		
205	0312		1 011 111 000		DSTC		FETCH STAT FLAGS FROM R17
206	0313		1 110 101 110		W,AXC		STAT FLAGS TO A
207	0314		0 010 101 000		CXM		HENRY'S FLAGS TO C
208	0315		1 011 110 000		DTDS		STORE HENRY'S FLAGS
209	0316		1 110 100 010		P,AXC		OPERATION MODE TO STAT FLAG
210	0317		1 110 101 110		W,AXC		NEW STAT FLAGS TO A AND M
211	0320		0 110 001 110		W,CTA		320
212	0321		0 010 101 000		CXM		DIGIT ENTRY IN B0, DATA IN D
213	0322		0 000 110 000		RETURN		
214	0323		0 100 001 100	OFST	PT4		OFFSET KEY
215	0324		1 101 111 010		XS,AM1A		
216	0325		1 101 111 010		XS,AM1A		IS THIS IN 1 VAR MODE?
217	0326	0107	0 100 011 111		BRN	NOTE	NO, PRINT NOTE 34
218	0327	0371	1 111 101 001		JSB	ADV	YES, PRINT "OF ="
219	0330	0366	1 111 011 101		JSB	BLANK	330
220	0331		1 010 001 100		PT10		
221	0332		0 010 011 000		LDC2		
222	0333	0366	1 111 011 101		JSB	BLANK	333
223	0334		0 010 001 100		PT2		
224	0335		0 000 011 000		LDC0		
225	0336		0 011 011 000		LDC3		
226	0337	0370	1 111 100 101		JSB	PRINT	
227	0340		0 010 101 000		CXM		340
228	0341		1 101 001 100		PT13		
229	0342		0 011 011 000		LDC3		SET FLAG FOR HISTOGRAM
230	0343		0 110 001 100		PT6		
231	0344		0 001 011 000		LDC1		SET FLAG FOR OFFSET ENTRY
232	0345		0 010 101 000	EC	CXM		

STATISTICS FUNCTION BLOCK LISTING (ROM Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
233	0346	0217	1 000 111 111		BRN FIN	
234	0347		1 100 010 000	DIGIT	ROM 6	DIGIT EGRESS TO ROM 6
235	0350		1 110 010 000	EXM712	ROM 7	350
236	0351		1 100 001 100	TESPRT	PT12	
237	0352		1 001 100 010		P,AM1	
238	0353	0057	0 010 111 111		BRN CONT	
239	0354	0074	0 011 110 011		BRN XPRINT	
240	0355		0 010 101 000	SUITE1	CXM	HENRYS FLAGS TO M
241	0356		1 001 001 100		PT9	RETURN STORED D REGISTER
242	0357	0361	1 111 001 001		JSB CH1A	
243	0360	0365	1 111 010 111		BRN SUITE2	
244	0361		1 000 000 100	CH0A	SS8	
245	0362		1 110 010 000	CH1A	ROM 7	
246	0363	0257	1 011 000 001		JSB PREP	DIGIT INTERPRETATION ENTRY
247	0364	0347	1 110 011 111		BRN DIGIT	
248	0365		0 100 101 000	SUITE2	CTS	
249	0366	0076	0 011 111 011		BRN SUITE3	
250	0367		1 110 010 000	BLANK	ROM 7	
251	0370		0 000 000 000		DUMMY	370
252	0371		1 000 000 100	PRINT	SS8	
253	0372		1 110 010 000	ADV	ROM 7	
254	0373		0 011 010 000	AK	TKRA	
255	0374	0217	1 000 111 111		BRN FIN	
256	0375	0107	0 100 011 111		BRN NOTE	
257	0376	0057	0 010 111 111		BRN CONT	
258	0377		0 000 000 000		DUMMY	377

STATISTICS FUNCTION BLOCK LISTING (ROM 1)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000	0166	0 111 011 011	PXACC	BRN	MINCN	0 INCREMENT N
4	0001		0 011 001 100		PT3		
5	0002	0161	0 111 000 111		BRN	MSTORE	STORE X IN R13
6	0003	0330	1 101 100 011		BRN	MSUM1	ADD TO SX
7	0004	0043	0 010 001 111		BRN	MSUM2	ADD TO SX2
8	0005	0203	1 000 001 111		BRN	MVAT1	IF NOT IN 2 VARIABLE MODE SKIP 2 STEPS
9	0006	0111	0 100 100 111		BRN	MSUM3	ADD TO SX4
10	0007	0114	0 100 110 011		BRN	MSUM4	ADD TO SX3
11	0010		0 011 001 100		PT3		10
12	0011	0163	0 111 001 111		BRN	MFCH	FETCH X
13	0012	0376	1 111 111 011		BRN	XPRINT	PRINT X
14	0013	0213	1 000 101 111		BRN	MTSDEL	CANCEL DELETE IF DATA SET IS COMPLETE
15	0014		1 010 010 000	XPLOT	ROM 5		TO X PLOT
16	0015	0255	1 010 111 001	MSUM5	JSB	FILREG	Y,Y,Y,Y
17	0016		0 100 001 100		PT4		
18	0017	0332	1 101 101 011		BRN	COMS	STORE SY Y, Y, Y, Y,
19	0020	0164	0 111 010 011	PYACC	BRN	MINCN2	20 INCREMENT N2
20	0021		0 100 001 100		PT4		
21	0022	0161	0 111 000 111		BRN	MSTORE	STORE Y IN R14
22	0023	0015	0 000 110 111		BRN	MSUM5	ADD TO SY
23	0024	0040	0 010 000 011		BRN	MSUM6	ADD TO SY2
24	0025	0205	1 000 010 111		BRN	MVAT2	IF NOT IN VARIABLE MODE SKIP 6 STEPS
25	0026	0055	0 010 110 111		BRN	MSUM7	
26	0027	0066	0 011 011 011		BRN	MSUM8	ADD TO SX2Y
27	0030	0223	1 001 001 111		BRN	MPLOT?	TEST FOR PLOT AND PEN RAISING
28	0031	0263	1 011 001 111		BRN	MLODCO	LOAD COORDINATES
29	0032	0275	1 011 110 111		BRN	MCHAR	PLOT POINT
30	0033	0250	1 010 100 011		BRN	XPENUP	RAISE PEN
31	0034		0 100 001 100		PT4		FETCH Y
32	0035	0163	0 111 001 111		BRN	MFCH	
33	0036	0376	1 111 111 011		BRN	XPRINT	PRINTY
34	0037	0213	1 000 101 111		BRN	MTSDEL	CANCEL DELETE
35	0040	0355	1 110 111 001	MSUM6	JSB	MULT	Y2, Y2, Y, Y
36	0041		0 101 001 100		PT5		40
37	0042	0332	1 101 101 011		BRN	COMS	STORE SY2 Y, Y, Y, Y
38	0043	0355	1 110 111 001	MSUM2	JSB	MULT	X2, X2, X, X
39	0044		0 100 101 000		CTS		
40	0045		1 100 101 000		DNR		X2, X2, X, X2
41	0046		0 010 001 100		PT2		
42	0047	0332	1 101 101 011		BRN	COMS	STORE SX2 X2, X2, X, X2
43	0050		1 011 110 000	SUITE2	DTDS		50 STORE NEW FREQ.
44	0051		0 001 001 100	BPXACC	PT1		LOAD STARTING ADDRESS FOR PXACC
45	0052		1 111 011 000		LDC15		
46	0053		1 111 011 000		LDC15		53
47	0054	0155	0 110 110 111		BRN	ROMID	
48	0055		0 011 001 100	MSUM7	PT3		
49	0056	0360	1 111 000 101		JSB	CHIA	X, Y, Y, Y
50	0057		1 100 101 000		DNR		
51	0060		0 110 101 000		STA		60
52	0061		0 110 101 000		STA		Y, X, X, X

(148)

STATISTICS FUNCTION BLOCK LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
53	0062	0355	1 110 111 001		JSB MULT	XY, XY
54	0063		0 100 101 000		CTS	XY, XY, XY, X
55	0064		0 011 001 100		PT3	
56	0065	0332	1 101 101 011		BRN COMS	STORE SXY X, X, XY, X
57	0066		1 100 101 000	MSUM8	DNR	X, XY, X, XY
58	0067	0355	1 110 111 001		JSB MULT	X2Y, X2Y
59	0070		1 000 001 100		PT8	70
60	0071	0332	1 101 101 011		BRN COMS	STORE SX2Y
61	0072		1 101 100 010		P,AM1A	IS PRGM IN ROM 1?
62	0073	0075	0 011 110 111		BRN R2	NO
63	0074		0 001 010 000		EERA	YES
64	0075		0 100 010 000	R2	ROM 2	
65	0076		0 010 101 000	TESTCOR	CXM	FLAG TO A
66	0077		0 110 001 110		W,CTA	
67	0100		0 010 101 000		CXM	100
68	0101		0 101 001 100		PT5	
69	0102		1 001 100 010		P,AM1	IS THIS IN DELETE MODE?
70	0103	0105	0 100 010 111		BRN COR	YES, SUBTRACT D FROM C
71	0104	0364	1 111 010 011		BRN PLUS	NO, ADD D AND C
72	0105		0 110 101 000	COR	STA	
73	0106		0 100 101 000		CTS	
74	0107		1 110 101 110		W,AXC	
75	0110	0363	1 111 001 111		BRN MINUS	110
76	0111	0355	1 110 111 001	MSUM3	JSB MULT	X4, X4, X, X2
77	0112		0 111 001 100		PT7	
78	0113	0332	1 101 101 011		BRN COMS	STORE SX4 X2, X2, X, X2
79	0114		1 100 101 000	MSUM4	DNR	X2, X, X2, X2
80	0115	0355	1 110 111 001		JSB MULT	X3, X3
81	0116		0 110 001 100		PT6	
82	0117	0332	1 101 101 011		BRN COMS	STORE SX3 X2, X2, X2, X2
83	0120	0344	1 110 010 101	MFREN	JSB DS	120 ACCESS TO MFREN
84	0121		0 100 101 000	CHALK	CTS	ACCESS TO CHALK FROM ROM 6
85	0122		0 100 101 000		CTS	I, I, I, DATA
86	0123		0 011 001 110		W,ZTC	
87	0124		0 111 101 110		W,CP1C	
88	0125		1 100 001 100		PT12	
89	0126		0 001 011 000		LDC1	
90	0127		0 011 011 000		LDC3	127 13, I, I, DATA
91	0130	0363	1 111 010 001		JSB PLUS	130 I+13, ", I, DATA
92	0131		1 100 001 100		PT12	
93	0132		0 101 100 010		P,CM1C	
94	0133		0 101 100 010		P,CM1C	IS CELL I IN CHIP 10 OR 11
95	0134	0137	0 101 111 111		BRN LD11	11
96	0135		1 010 011 000		LDC10	10
97	0136	0140	0 110 000 011		BRN F	
98	0137		1 011 011 000	LD11	LDC11	
99	0140		1 001 110 000	F	ATDS	140 ADDRESS CELL I
100	0141		0 110 101 000		STA	
101	0142		1 011 111 000		DSTC	FREQ(I), I, DATA, DATA
102	0143		1 001 010 100		YS9	SHOULD FREQ BE INCREMENTED?
103	0144	0375	1 111 110 111		BRN CONT	NO
104	0145		1 100 001 100		PT12	YES
105	0146		0 110 001 110		W,CTA	
106	0147		0 011 001 110		W,ZTC	
107	0150		0 001 011 000		LDC1	150
108	0151	0316	1 100 111 011		BRN SUITE1	
109	0152		0 001 001 100	BPYACC	PT1	LOAD STARTING ADDRESS FOR PYACC PRGM
110	0153		0 000 011 000		LDC0	
111	0154		1 111 011 000		LDC15	
112	0155		0 010 101 000	ROMID	CXM	

STATISTICS FUNCTION BLOCK LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 111 001 100		PT7		
114	0157		0 000 011 000		LDC0		
115	0160		0 000 010 000		ROM 0		160
116	0161		0 100 010 000	MSTORE	ROM 2		
117	0162		0 000 000 000		DUMMY		
118	0163		0 100 010 000	MFCH	ROM 2		
119	0164		1 001 001 100	MINCN2	PT9		—, DATA, —, —
120	0165	0167	0 111 011 111		BRN	CIN	
121	0166		0 000 001 100	MINCN	PT0		
122	0167	0357	1 111 000 001	CIN	JSB	CH0A	N, DATA
123	0170		1 110 101 110		W,AXC		170
124	0171		0 011 001 110		W,ZTC		
125	0172		1 100 001 100		PT12		
126	0173		0 001 011 000		LDC1		
127	0174		0 100 101 000		CTS		1, 1, DATA, —
128	0175		1 110 101 110		W,AXC		N, 1, DATA
129	0176	0075	0 011 111 001		JSB	TESTCOR	NEW N, ", DATA
130	0177		1 011 110 000		DTDS		
131	0200		1 100 101 000		DNR		200
132	0201		1 100 101 000		DNR		
133	0202	0375	1 111 110 111		BRN	CONT	
134	0203		0 010 001 100	MVAT1	PT2		203
135	0204	0206	1 000 011 011		BRN	CVT	
136	0205		0 110 001 100	MVAT2	PT6		
137	0206		1 101 111 010	CVT	XS,AM1A		
138	0207		1 101 111 010		XS,AM1A		
139	0210		1 101 111 010		XS,AM1A		210 IS THIS IN 2 VAR MODE?
140	0211	0347	1 110 011 111		BRN	SKIP*	NO
141	0212	0375	1 111 110 111		BRN	CONT	YES
142	0213		0 011 001 100	MTSDEL	PT3		
143	0214		1 001 100 010		P,AM1		IS DATA SET COMPLETE?
144	0215	0373	1 111 101 111		BRN	FIN	NO
145	0216		0 010 101 000		CXM		YES, CANCEL DELETE, 0 --> F5
146	0217		0 101 001 100		PT5		
147	0220		0 011 000 010		P,ZTC		220
148	0221		0 010 101 000		CXM		
149	0222	0373	1 111 101 111		BRN	FIN	
150	0223		1 000 001 100	MPLOT?	PT8		
151	0224		1 001 100 010		P,AM1		IS THIS IN PLOTTING MODE?
152	0225	0230	1 001 100 011		BRN	TIN	YES
153	0226		0 011 001 100		PT3		NO
154	0227	0347	1 110 011 111		BRN	SKIP*	SKIP TO PRINT
155	0230		0 101 001 100	TIN	PT5		230
156	0231		1 101 100 010		P,AM1A		IS THIS IN DELETE MODE?
157	0232	0250	1 010 100 011		BRN	XPENUP	YES, RAISE PEN
158	0233		0 000 001 100		PT0		NO
159	0234		0 101 011 000		LDC5		
160	0235		0 000 001 100		PT0		
161	0236		0 001 000 010		P,AMC		IS PLOT TO BE CONNECTED?
162	0237	0241	1 010 000 111		BRN	NT	YES, TEST N
163	0240	0250	1 010 100 011		BRN	XPENUP	240 NO, RAISE PEN
164	0241	0357	1 111 000 001	NT	JSB	CH0A	FETCH N
165	0242		1 011 101 110		W,ZTA		
166	0243		1 100 001 100		PT12		
167	0244		1 111 100 010		P,AP1A		
168	0245		0 101 001 110		W,AMCC		N–1 TO C
169	0246		0 001 101 110		W,CM1		IS N=1?
170	0247	0014	0 000 110 011		BRN	XPLOT	NO, DROP PEN
171	0250		1 011 001 100	XPENUP	PT11	250,	YES, RAISE PEN
172	0251	0014	0 000 110 011		BRN	XPLOT	

STATISTICS FUNCTION BLOCK LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 000 000 000		DUMMY		
174	0253		0 000 000 000		DUMMY		
175	0254		0 000 000 000		DUMMY		
176	0255		0 000 000 000		DUMMY		
177	0256		0 100 101 000	FILREG	CTS		
178	0257		0 100 101 000		CTS		257
179	0260		0 100 101 000		CTS		260
180	0261		0 001 100 100		RS1		RETURN FROM ROM 7 ROUTINES
181	0262		0 000 110 000		RETURN		
182	0263		0 011 001 100	MLODCO	PT3		FETCH X DATA
183	0264	0360	1 111 000 101		JSB	CH1A	
184	0265		0 100 101 000		CTS		
185	0266	0344	1 110 010 101		JSB	DS	ADDRESS DEDICATES STORAGE
186	0267		1 100 101 000		DNR		
187	0270		1 011 110 000		DTDS		270 STORE X DATA IN DS
188	0271		0 100 001 100		PT4		FETCH Y DATA
189	0272	0360	1 111 000 101		JSB	CH1A	
190	0273		1 010 001 100		PT10		
191	0274	0014	0 000 110 011		BRN	XPLOT	SEND PEN TO NEW COORDINATES
192	0275		0 101 001 100	MCHAR	PT5		
193	0276		1 001 100 010		P,AM1		IS THIS IN DELETE MODE?
194	0277	0312	1 100 101 011		BRN	SCRATCH	YES, SET UP SCRATCH CHARACTER
195	0300		0 000 001 100		PT0		300 N, SET UP SELECTED CHARACTER
196	0301		0 101 011 000		LDC5		
197	0302		0 000 001 100		PT0		
198	0303		1 111 000 010		P,APCA		IS PLOT CONNECTED
199	0304	0306	1 100 011 011		BRN	ATC	YES,
200	0305		1 101 000 010		P,AMCA		NO, RESTORE PLOT CHARACTER
201	0306		1 110 100 010	ATC	P,AXC		CHARACTER TO C
202	0307		1 011 101 110	CTA	W,ZTA		ARRANGE A FOR XPLOT
203	0310		1 110 100 010		P,AXC		310
204	0311	0014	0 000 110 011		BRN	XPLOT	PLOT CHARACTER
205	0312		0 000 001 100	SCRATCH	PT0		
206	0313		1 001 011 000		LDC9		
207	0314		0 000 001 100		PT0		
208	0315	0307	1 100 011 111		BRN	CTA	
209	0316		0 100 101 000	SUITE1	CTS		
210	0317		1 110 101 110		W,AXC		FREQ, I, I, DATA
211	0320	0075	0 011 111 001		JSB	TESTCOR	320 NEW FREQ, ", I, DATA
212	0321		0 110 101 000		STA		NEW FREQ, I, DATA, DATA
213	0322		0 110 101 000		STA		NEW FREQ, DATA, DATA, DATA
214	0323		1 001 100 100		RS9		
215	0324		0 101 001 100		PT5		
216	0325		0 001 111 110		S,CM1		WAS CELL COUNT 0?
217	0326	0374	1 111 110 011		BRN	NOTE	YES, NOTE 35 NO DELETION
218	0327	0050	0 010 100 011		BRN	SUITE2	NO
219	0330	0255	1 010 111 001	MSUM1	JSB	FILREG	330 X, X, X, X
220	0331		0 001 001 100		PT1		
221	0332	0357	1 111 000 001	COMS	JSB	CH0A	SX, X, X, X
222	0333	0075	0 011 111 001		JSB	TESTCOR	333
223	0334		1 011 110 000		DTDS		D1, D2, D3, D4
224	0335		0 110 101 000		STA		
225	0336		0 110 101 000		STA		D1, D4, D4, D4
226	0337		1 110 101 110		W,AXC		D3
227	0340		0 110 101 000		STA		340
228	0341		0 100 101 000		CTS		D3, D3, D4, D4
229	0342		1 110 101 110		W,AXC		D4
230	0343		0 100 101 000		CTS		D4, D4, D3, D4
231	0344	0375	1 111 110 111		BRN	CONT	
232	0345		0 001 000 100	DS	SS1		

STATISTICS FUNCTION BLOCK LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		1 110 010 000		ROM 7		
234	0347		0 111 000 000	SKIP+	PINC		
235	0350		0 111 000 000		PINC		350
236	0351		0 000 011 100		PRS		
237	0352		0 000 101 100		YP0		
238	0353	0347	1 110 011 111		BRN	SKIP+	
239	0354	0375	1 111 110 111		BRN	CONT	
240	0355		0 000 000 000		DUMMY		
241	0356		0 001 000 100	MULT	SS1		
242	0357		1 110 010 000		ROM 7		
243	0360		1 000 000 100	CH0A	SS8		360
244	0361		0 001 000 100	CH1A	SS1		
245	0362		1 110 010 000		ROM 7		
246	0363		0 011 111 110	MINUS	S,ZNCC		
247	0364		0 001 000 100	PLUS	SS1		
248	0365		1 110 010 000		ROM 7		
249	0366		0 000 000 000		DUMMY		
250	0367		0 000 000 000		DUMMY		
251	0370		0 000 000 000		DUMMY		370
252	0371		0 000 000 000		DUMMY		
253	0372		0 000 000 000		DUMMY		
254	0373		0 000 010 000	FIN	ROM 0		
255	0374		0 000 010 000	NOTE	ROM 0		
256	0375		0 000 010 000	CONT	ROM 0		
257	0376		0 100 010 000	XPRINT	ROM 2		
258	0377	0347	1 110 011 111		BRN	SKIP+	377 ACCESS TO SKIP+

STATISTICS FUNCTION BLOCK LISTING (ROM 2)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		1 010 010 000	XROOT	ROM 5			0	
4	0001	0127	0 101 011 111	PBAS	BRN	MLDN	FETCH N AND ADVANCE PAPER		
5	0002	0174	0 111 110 011		BRN	MLDEQ	PRINT "NI="		
6	0003	0227	1 001 011 111		BRN	MPRINTN1			
7	0004	0377	1 111 111 111		BRN	XPRINT	PRINT NI		
8	0005	0116	0 100 111 011		BRN	MEANX	CALC. THE MEAN OF X		
9	0006		0 011 001 100		PT3				
10	0007	0167	0 111 011 111		BRN	MSTORE	STORE MEAN OF X IN R13		
11	0010	0174	0 111 110 011		BRN	MLDEQ	10 PRINT "XM="		
12	0011	0222	1 001 001 011		BRN	MPRINTXM			
13	0012	0377	1 111 111 111		BRN	XPRINT	PRINT MEAN OF X		
14	0013	0276	1 011 111 011		BRN	MSTX	UX, UX		
15	0014	0133	0 101 101 111		BRN	MCAL1	UX(N–1)."		
16	0015	0000	0 000 000 011		BRN	XROOT	SDX,		
17	0016		0 100 001 100		PT4				
18	0017	0167	0 111 011 111		BRN	MSTORE	STORE SDX IN R14		
19	0020	0174	0 111 110 011		BRN	MLDEQ	20 PRINT "SI="		
20	0021	0214	1 000 110 011		BRN	MPRINTS1			
21	0022	0377	1 111 111 111		BRN	XPRINT	PRINT SDX		
22	0023	0203	1 000 001 111		BRN	MTV1	IF 2 SAMPLE, CONT; IF 1 VAR, FIN; IF 2 VAR, SKIP 5		
23	0024	0102	0 100 001 011		BRN	MEXNS	EXCHANGE N1 AND N2 IN DATA STORAGE		
24	0025	0127	0 101 011 111		BRN	MLDN	FETCH N2 AND ADVANCE PAPER		
25	0026	0174	0 111 110 011		BRN	MLDEQ	PRINT "N2 ="		
26	0027	0350	1 110 100 011		BRN	MPRINTN2			
27	0030	0377	1 111 111 111		BRN	XPRINT	30 PRINT N2		
28	0031	0120	0 101 000 011		BRN	MEANY	CALC. THE MEAN OF Y		
29	0032		0 000 001 100		PT0				
30	0033	0167	0 111 011 111		BRN	MSTORE	STORE MEAN OF Y IN R10		
31	0034	0174	0 111 110 011		BRN	MLDEQ	PRINT "YM ="		
32	0035	0225	1 001 010 111		BRN	MPRINTYM			
33	0036	0377	1 111 111 111		BRN	XPRINT	PRINT MEAN OF Y		
34	0037	0275	1 011 110 111		BRN	MSTY	UY, UY		
35	0040	0133	0 101 101 111		BRN	MCAL1	40 UY/(N–1)		
36	0041	0000	0 000 000 011		BRN	XROOT	SDY		
37	0042		0 001 001 100		PT1				
38	0043	0167	0 111 011 111		BRN	MSTORE	STORE SDY IN R11		
39	0044	0174	0 111 110 011		BRN	MLDEQ	PRINT "S2 ="		
40	0045	0217	1 000 111 111		BRN	MPRINTS2			
41	0046	0377	1 111 111 111		BRN	XPRINT	PRINT SDY		
42	0047	0205	1 000 010 111		BRN	MTV2	IF 2 SAMPLE, CONT; IF 2 VAR SKIP2		
43	0050	0102	0 100 001 011		BRN	MEXNS	50 EXCHANGE N1 AND N2 IN DATA STORAGE		
44	0051	0252	1 010 101 011		BRN	BOUT			
45	0052	0276	1 011 111 011		BRN	MSTX	UX, UX		
46	0053	0275	1 011 110 111		BRN	MSTY	53 UY, UY, UX		
47	0054	0145	0 110 010 111		BRN	MCAL2	UYUX, "		
48	0055	0000	0 000 000 011		BRN	XROOT	R(UXUY)		
49	0056	0273	1 011 101 111		BRN	MLUKXY	UXY,UXY,R(UXUY)		
50	0057	0277	1 011 111 111		BRN	MCAL3	R,R		
51	0060		0 010 001 100		PT2		60		
52	0061	0167	0 111 011 111		BRN	MSTORE	STORE R IN R12		

STATISTICS FUNCTION BLOCK LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0174	0 111 110 011		BRN	MLDFQ	PRINT "R="
54	0063	0234	1 001 110 011		BRN	MPRINTR	
55	0064	0377	1 111 111 111		BRN	XPRINT	PRINT R
56	0065	0252	1 010 101 011		BRN	BOUT	
57	0066		0 001 001 100	WIDTH	PT1		LOAD STARTING ADDRESS FOR PRTW PRGM
58	0067		0 011 011 000		LDC3		
59	0070		1 001 011 000		LDC9		70
60	0071	0155	0 110 110 111		BRN	ROMID	
61	0072	0326	1 101 011 011	PRTW	BRN	MDPRT	PRINT CELL WIDTH
62	0073		0 101 001 100		PT5		
63	0074	0167	0 111 011 111		BRN	MSTORE	STORE CELL WIDTH IN R15
64	0075	0252	1 010 101 011		BRN	BOUT	
65	0076		1 101 100 010		P,AMIA		IS PRGM IN ROM 2?
66	0077	0101	0 100 000 111		BRN	R3	NO
67	0100		0 001 010 000		EFRA		100 YES
68	0101		0 110 010 000	R3	ROM3		LOOK FOR PRGM IN ROM 3
69	0102		1 001 001 100	MFXNS	PT9		FETCH N2
70	0103	0357	1 111 000 001		JSB	CH0A	
71	0104		1 110 101 110		W,AXC		N2 TO A
72	0105		0 000 001 100		PT0		FETCH N1
73	0106	0357	1 111 000 001		JSB	CH0A	
74	0107		1 110 101 110		W,AXC		N1 TO A, N2 TO C
75	0110		1 011 110 000		DTDS		110 STORE N2 IN R00
76	0111		1 001 001 100		PT9		ADDRESS R09
77	0112	0357	1 111 000 001		JSB	CH0A	
78	0113		1 110 101 110		W,AXC		N1 TO C
79	0114		1 011 110 000		DTDS		STORE N1 IN R09
80	0115	0375	1 111 110 111		BRN	CONT	
81	0116		0 001 001 100	MFANX	PT1		
82	0117	0101	0 101 000 111		BRN	COM21	
83	0120		0 100 001 100	MFANY	PT4		120
84	0121	0357	1 111 000 001	COM21	JSB	CH0A	SX (ALSO FOR Y)
85	0122		0 100 101 000		CTS		SX, SX
86	0123		0 000 001 100		PT0		
87	0124	0357	1 111 000 001		JSB	CH0A	N, SX
88	0125	0247	1 010 100 001		JSB	DIVIDE	SX/N,"
89	0126	0375	1 111 110 111		BRN	CONT	
90	0127		0 000 001 100	MLDN	PT0		127
91	0130	0357	1 111 000 001		JSB	CH0A	130 N--> C
92	0131	0370	1 111 100 101		JSB	ADV	ADVANCE PAPER
93	0132	0375	1 111 110 111		BRN	CONT	
94	0133		0 000 001 100	MCALI	PT0		UX,UX
95	0134	0357	1 111 000 001		JSB	CH0A	N, UX
96	0135		0 100 101 000		CTS		N, N, UX
97	0136		0 011 001 110		W,2TC		
98	0137		1 100 001 100		PT12		
99	0140		0 001 011 000		LDC1		140 1, N, UX
100	0141	0362	1 111 001 101		JSB	MINUS	N–1, N–1, UX
101	0142		1 100 101 000		DNR		N–, UX
102	0143	0247	1 010 100 001	COD	JSB	DIVIDE	UX/(N–1),"
103	0144	0375	1 111 110 111		BRN	CONT	
104	0145		0 110 101 000	MCAL2	STA		UY, UX
105	0146	0355	1 110 111 001		JSB	MULT	UYUX,"
106	0147	0375	1 111 110 111		BRN	CONT	
107	0150		1 101 100 010		P,AM1A		150 ACCESS TO OFFSET/WIDTH ENTRIES
108	0151	0066	0 011 011 011		BRN	WIDTH	
109	0152		0 001 001 100	BPRTO	PT1		LOAD STARTING ADDRESS FOR PRTO PRGM
110	0153		1 100 011 000		LDC12		
111	0154		0 010 011 000		LDC2		
112	0155		0 010 101 000	ROMID	CXM		

STATISTICS FUNCTION BLOCK LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 111 001 100		PT7		
114	0157		0 001 011 000		LDC1		
115	0160		0 000 010 000		ROM 0		160
116	0161		0 000 000 000		DUMMY		
117	0162	0167	0 111 011 111		BRN	MSTORE	MSTORE ACCESS
118	0163	0174	0 111 110 011		BRN	MLDEQ	MLDEQ ACCESS
119	0164	0360	1 111 000 101		JSB	CH1A	MFCH ACCESS
120	0165		0 001 001 100	FCOM	PT1		SKIP 1 STEP DUE TO POINTER IN PRGM
121	0166	0376	1 111 111 011		BRN	SKIP+	
122	0167		0 100 101 000	MSTORE	CTS		
123	0170	0360	1 111 000 101		JSB	CH1A	170 ADDRESS REGISTER CORRESPONDING TO POINTER
124	0171		1 100 101 000		DNR		
125	0172		1 011 110 000		DTDS		STORE DATA ORIGINALLY IN C
126	0173	0165	0 111 010 111		BRN	FCOM	
127	0174		0 100 101 000	MLDEQ	CTS		
128	0175	0370	1 111 100 101		JSB	ADV	ADVANCE PAPER
129	0176	0365	1 111 011 001		JSB	BLANK	LOAD BLANKS IN C
130	0177		1 010 001 100		PT10		LOAD "="
131	0200		0 010 011 000		LDC2		200
132	0201	0365	1 111 011 001		JSB	BLANK	TRANSFER C TO T AND BLANK C
133	0202	0375	1 111 110 111		BRN	CONT	
134	0203		0 101 001 100	MTV1	PT5		203
135	0204	0206	1 000 011 011		BRN	T1	
136	0205		0 010 001 100	MTV2	PT2		
137	0206		1 101 111 010	T1	XS,AM1A		WHICH DATA ENTRY MODE?
138	0207	0211	1 000 100 111		BRN	T2	
139	0210	0375	1 111 110 111		BRN	CONT	210 2 SAMPLE
140	0211		1 101 111 010	T2	XS,AM1A		
141	0212	0376	1 111 111 011		BRN	SKIP+	2 VARIABLE
142	0213	0252	1 010 101 011		BRN	BOUT	
143	0214		1 101 001 100	MPRINTS1	PT13		
144	0215		0 110 011 000		LDC6		
145	0216	0231	1 001 100 111		BRN	LD1	
146	0217		1 101 001 100	MPRINTS2	PT13		
147	0220		0 110 011 000		LDC6		220
148	0221	0352	1 110 101 011		BRN	LD2	
149	0222		1 101 001 100	MPRINTXM	PT13		
150	0223		0 000 011 000		LDC0		
151	0224	0236	1 001 111 011		BRN	COMPT	
152	0225		1 101 001 100	MPRINTYM	PT13		
153	0226	0231	1 001 100 111		BRN	LD1	
154	0227		1 011 001 100	MPRINTN1	PT11		
155	0230		1 100 011 000		LDC12		230
156	0231		0 001 011 000	LD1	LDC1		
157	0232	0236	1 001 111 011		BRN	COMPT	
158	0233		0 000 000 000		DUMMY		
159	0234		1 101 001 100	MPRINTR	PT13		
160	0235		1 011 011 000		LDC11		
161	0236	0367	1 111 100 001	COMPT	JSB	PRINT	
162	0237		1 100 101 000		DNR		
163	0240	0375	1 111 110 111		BRN	CONT	240
164	0241	-	0 011 001 100	BASIC	PT3		BASIC STAT KEY
165	0242		1 001 100 010		P,AM1		IS DATA SET COMPLETE?
166	0243	0374	1 111 110 011		BRN	NOTE	NO, PRINT NOTE 33
167	0244		0 001 001 100	BPBAS	PT1		YES, LOAD STARTING ADDRESS FOR PBAS PRGM.
168	0245		0 000 011 000		LDC0		
169	0246		0 000 011 000		LDC0		
170	0247	0155	0 110 110 111		BRN	ROMID	
171	0250		0 010 000 100	DIVIDE	SS2		250
172	0251	0355	1 110 110 111		BRN	DIV1	

STATISTICS FUNCTION BLOCK LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
173	0252	0370	1 111 100 101	BOUT	JSB	ADV			
174	0253	0370	1 111 100 101		JSB	ADV			
175	0254	0370	1 111 100 101		JSB	ADV			
176	0255	0370	1 111 100 101		JSB	ADV			
177	0256	0370	1 111 100 101		JSB	ADV			
178	0257	0266	1 011 011 011		BRN	SUITE1			
179	0260		0 000 000 000		DUMMY				
180	0261		0 000 000 000		DUMMY				
181	0262		0 001 001 100	LINJ	PT1		ACCESS TO LINJ		
182	0263	0330	1 101 100 011		BRN	SUITE2			
183	0264		0 010 100 100		RS2		RETURN FROM ROM 7 ROUTINES		
184	0265		0 000 110 000		RETURN				
185	0266		0 010 101 000	SUITE1	CXM		SET F6 FOR VARIABLE ENTRY		
186	0267		0 110 001 100		PT 6				
187	0270		0 000 011 000		LDC0		270		
188	0271		0 010 101 000		CXM				
189	0272	0373	1 111 101 111		BRN	FIN			
190	0273		0 100 101 000	MLUKXY	CTS				
191	0274		0 110 010 000	MSTXY	ROM 3		EGRESS TO MSTXY		
192	0275		0 110 010 000	MSTY	ROM 3		EGRESS TO MSTY		
193	0276		0 110 010 000	MSTX	ROM 3		EGRESS TO MSTX		
194	0277		1 100 101 000	MCAL 3	DNR		UXY, R(UXUY), –, UXY		
195	0300		1 100 101 000		DNR		300 R(UXUY)--, UXY, UXY		
196	0301		0 110 101 000		STA		R(UXUY), UXY		
197	0302	0143	0 110 001 111		BRN	COD	UXY/R(UXUY)=R		
198	0303	0326	1 101 011 011	PRTO	BRN	MDPRT	PRINT OFFSET		
199	0304		0 110 001 100		PT6				
200	0305	0167	0 111 011 111		BRN	MSTORE	STORE OFFSET IN R16		
201	0306	0370	1 111 100 101		JSB	ADV	ADVANCE PAPER		
202	0307	0365	1 111 011 001		JSB	BLANK	PRINT "CEL="		
203	0310		1 101 001 100		PT13		310		
204	0311		1 011 011 000		LDC11				
205	0312		1 011 001 100		PT11				
206	0313		1 001 011 000		LDC9				
207	0314		0 010 011 000		LDC2				
208	0315	0365	1 111 011 001		JSB	BLANK			
209	0316		0 001 001 100		PT1				
210	0317		0 001 011 000		LDC1				
211	0320	0367	1 111 100 001		JSB	PRINT	320		
212	0321		0 010 101 000		CXM		SET F6 FOR CELL WIDTH ENTRY		
213	0322		0 110 001 100		PT6				
214	0323		0 010 011 000		LDC2				
215	0324		0 010 101 000		CXM				
216	0325	0373	1 111 101 111		BRN	FIN			
217	0326		1 100 101 000	MDPRT	DNR				
218	0327	0377	1 111 111 111		BRN	XPRINT			
219	0330	0360	1 111 000 101	SUITE2	JSB	CH1A	330 B, CSX2, –, –		
220	0331		0 100 101 000		CTS		B, B, CSX2, –		
221	0332		0 001 001 100		PT1				
222	0333	0357	1 111 000 001		JSB	CH0A	333 SX, B, CSX2, –		
223	0334	0355	1 110 111 001		JSB	MULT	BSX, BSX, CSX2, –		
224	0335		1 100 101 000		DNR		BSX, CSX2, –, BSX		
225	0336	0363	1 111 010 001		JSB	PLUS	BSX+CSX2,"		
226	0337		0 100 001 100		PT4				
227	0340	0357	1 111 000 001		JSB	CH0A	340 SY,BSX+CSX2		
228	0341	0362	1 111 001 101		JSB	MINUS	BSX+CSX2–SY,"		
229	0342		0 000 001 100		PT0				
230	0343	0357	1 111 000 001		JSB	CH0A	N		
231	0344		0 011 111 110		S,ZNCC		–N		
232	0345	0247	1 010 100 001		JSB	DIVIDE	(SY–DSX–CSX2)/N- A		

STATISTICS FUNCTION BLOCK LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 000 001 100		PT0		
234	0347	0167	0 111 011 111		BRN	MSTORE	STORE A IN R10
235	0350		1 011 001 100	MPRINTN2	PT11		
236	0351		1 100 011 000		LDC12		
237	0352		0 010 011 000	LD2	LDC2		
238	0353	0236	1 001 111 011		BRN	COMPT	
239	0354	0252	1 010 101 011		BRN	BOUT	ACCESS TO BOUT
240	0355		1 110 010 000	DIV1.	ROM 7		
241	0356		0 010 000 100	MULT	SS2		
242	0357		1 110 010 000		ROM 7		
243	0360		1 000 000 100	CH0A	SS8		360
244	0361		0 010 000 100	CH1A	SS2		
245	0362		1 110 010 000		ROM 7		
246	0363		0 011 111 110	MINUS	S,ZNCC		
247	0364		0 010 000 100	PLUS	SS2		
248	0365		1 110 010 000		ROM 7		
249	0366		0 010 000 100	BLANK	SS2		
250	0367		1 110 010 000		ROM 7		
251	0370		1 000 000 100	PRINT	SS8		370
252	0371		0 010 000 100	ADV	SS2		
253	0372		1 110 010 000		ROM 7		
254	0373		0 000 010 000	FIN	ROM 0		
255	0374		0 000 010 000	NOTE	ROM 0		
256	0375		0 000 010 000	CONT	ROM 0		
257	0376		0 010 010 000	SKIP+	ROM 1		
258	0377		1 010 010 000	XPRINT	ROM 5		377

STATISTICS FUNCTION BLOCK LISTING (ROM 3)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		1 100 101 000	MCAL10	DNR		0		
4	0001	0355	1 110 111 001	MCAL11	JSB	MULT			
5	0002	0375	1 111 110 111		BRN	CONT			
6	0003		1 100 101 000	MCAL12	DNR				
7	0004	0362	1 111 001 101		JSB	MINUS			
8	0005	0375	1 111 110 111		BRN	CONT			
9	0006		1 100 101 000	MCAL15	DNR				
10	0007	0353	1 110 110 001	MCAL17	JSB	DIVIDE			
11	0010	0375	1 111 110 111		BRN	CONT	10		
12	0011		1 100 101 000	MCAL19	DNR				
13	0012	0363	1 111 010 001	MCAL16	JSB	PLUS			
14	0013	0375	1 111 110 111		BRN	CONT			
15	0014		0 001 001 100	MPRINTA	PT1				
16	0015		1 001 011 000		LDC9				
17	0016	0100	0 100 000 011		BRN	PC			
18	0017		0 001 001 100	MPRINTB	PT1				
19	0020		0 000 011 000		LDC0		20		
20	0021	0100	0 100 000 011		BRN	PC			
21	0022	0262	1 011 001 101		JSB	V2?	LINEAR KEY		
22	0023		0 011 001 100		PT3				
23	0024		1 001 100 010		P,AM1				
24	0025	0374	1 111 110 011		BRN	NOTE			
25	0026		0 010 101 000		CXM		SET F13 TO INDICATE LINEAR REG.		
26	0027		0 011 011 110		S,ZTC				
27	0030		0 111 111 110		S,CP1C		30		
28	0031	0072	0 011 101 011		BRN	SUITE2			
29	0032	0262	1 011 001 101		JSB	V2?	PAR KEY		
30	0033		0 011 001 100		PT3				
31	0034		1 001 100 010		P,AM1				
32	0035	0374	1 111 110 011		BRN	NOTE			
33	0036		0 010 101 000		CXM				
34	0037		0 011 011 110		S,ZTC				
35	0040		0 111 111 110		S,CP1C		40		
36	0041		0 111 111 110		S,CP1C				
37	0042		0 010 101 000		CXM				
38	0043		0 001 001 100	BPARA	PT1		LOAD STARTING ADDRESS FOR PARA PRGM		
39	0044		0 111 011 000		LDC7				
40	0045		0 011 011 000		LDC3				
41	0046	0155	0 110 110 111		BRN	ROMID			
42	0047	0326	1 101 011 011	PLIN	BRN	MSTXY	LIN.REG.PRGM	UXY, UXY	
43	0050	0277	1 011 111 111		BRN	MSTX	50	UX, UX, UXY	
44	0051	0006	0 000 011 011		BRN	MCAL15		UXY/UX=B	
45	0052		0 001 001 100		PT1				
46	0053	0161	0 111 000 111		BRN	MSTORE	53	STORE B IN R11	
47	0054	0253	1 010 101 111		BRN	MCALA		CALC A AND STORE R10	
48	0055		0 000 000 000		DUMMY				
49	0056	0326	1 101 011 011		BRN	MSTXY		UXY,UXY	
50	0057	0001	0 000 000 111		BRN	MCAL11		(UXY)2, "	
51	0060	0277	1 011 111 111		BRN	MSTX	60	UX, UX, (UXY)2	
52	0061	0006	0 000 011 011		BRN	MCAL15		(UXY)2/UX	

STATISTICS FUNCTION BLOCK LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0062	0316	1 100 111 011		BRN	MSTY	UY, UY, (UXY)2/UX	
54	0063	0006	0 000 011 011		BRN	MCAL15	(UXY)2/(UXUY)=R2	
55	0064		0 011 001 100		PT3			
56	0065	0161	0 111 000 111		BRN	MSTORE	STORE R2 IN R13	
57	0066		0 001 001 100	BPRINTS	PT1		LOAD STARTING ADDRESS FOR PRINTS PRGM	
58	0067		0 100 011 000		LDC4			
59	0070		1 100 011 000		LDC12		70	
60	0071	0155	0 110 110 111		BRN	ROMID		
61	0072		0 010 101 000	SUITE2	CXM			
62	0073		0 010 001 100		PT2		SET R12(C) TO 0	
63	0074	0360	1 111 000 101		JSB	CH1A		
64	0075	0150	0 110 100 011		BRN	BPLIN		
65	0076		1 101 001 100	MPRINTR2	PT13			
66	0077		1 000 011 000		LDC8			
67	0100	0367	1 111 100 001	PC	JSB	PRINT	100	
68	0101	0271	1 011 100 111		BRN	SUITE1		
69	0102		1 101 100 010		P,AM1A		IS PRGM IN ROM 3?	
70	0103	0105	0 100 010 111		BRN	R4	NO, LOOK IN ROM 4	
71	0104		0 001 010 000		EERA		YES, GO TO NEXT STEP	
72	0105		1 000 010 000	R4	ROM 4			
73	0106	0355	1 110 111 001	STAT	JSB	MULT	SXSY, SXSY, SXY (UXY EXAMPLE)	
74	0107		0 000 001 100		PT0			
75	0110	0357	1 111 000 001		JSB	CH0A	110 N, SXSY, SXY	
76	0111	0353	1 110 110 001		JSB	DIVIDE	SXSY/N, ", SXY	
77	0112		1 100 101 000		DNR			
78	0113	0362	1 111 001 101		JSB	MINUS	SXY–SXSY/N=UXY	
79	0114	0375	1 111 110 111		BRN	CONT		
80	0115		0 000 001 100	PRINTS	PT0		FETCH A	
81	0116	0163	0 111 001 111		BRN	MFCH		
82	0117	0162	0 111 001 011		BRN	MLDEQ	PRINT "A ="	
83	0120	0014	0 000 110 011		BRN	MPRINTA	120	
84	0121	0377	1 111 111 111		BRN	XPRINT	PRINT A	
85	0122		0 001 001 100		PT1		FETCH B	
86	0123	0163	0 111 001 111		BRN	MFCH		
87	0124	0162	0 111 001 011		BRN	MLDEQ	PRINT "B ="	
88	0125	0017	0 000 111 111		BRN	MPRINTB		
89	0126	0377	1 111 111 111		BRN	XPRINT	PRINT B	
90	0127	0143	0 110 001 111		BRN	MTR	127 SKIP C PRINTOUT IF LINEAR	
91	0130		0 010 001 100		PT2		130 FETCH C	
92	0131	0163	0 111 001 111		BRN	MFCH		
93	0132	0162	0 111 001 011		BRN	MLDEQ	PRINT "C ="	
94	0133	0350	1 110 100 011		BRN	MPRINTC		
95	0134	0377	1 111 111 111		BRN	XPRINT	PRINT C	
96	0135		0 011 001 100		PT3		FETCH R2	
97	0136	0163	0 111 001 111		BRN	MFCH		
98	0137	0162	0 111 001 011		BRN	MLDEQ	PRINT "R2 ="	
99	0140	0076	0 011 111 011		BRN	MPRINTR2	140	
100	0141	0377	1 111 111 111		BRN	XPRINT	PRINT R2	
101	0142	0353	1 110 101 111		BRN	BOUT		
102	0143		0 101 001 100	MTR	PT5			
103	0144		1 101 111 110		S,AM1A			
104	0145		1 101 111 110		S,AM1A		IS THIS A LINEAR REGRESSION?	
105	0146	0375	1 111 110 111		BRN	CONT	NO	
106	0147	0376	1 111 111 011		BRN	SKIP+	YES	
107	0150		0 011 001 110	BPLIN	W,ZTC		150	
108	0151		1 011 110 000		DTDS			
109	0152		0 001 001 100		PT1		LOAD STARTING ADDRESS FOR PLIN PRGM	
110	0153		0 010 011 000		LDC2			
111	0154		0 110 011 000		LDC6			
112	0155		0 010 101 000	ROMID	CXM		LOAD ROM IDENTIFICATION FLAG	

STATISTICS FUNCTION BLOCK LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156		0 111 001 100		PT7			
114	0157		0 010 011 000		LDC2			
115	0160		0 000 010 000		ROM 0		160	
116	0161		0 100 010 000	MSTORE	ROM 2		EGRESS TO MSTORE	
117	0162		0 100 010 000	MLDEQ	ROM 2		EGRESS TO MLDEQ	
118	0163		0 100 010 000	MFCH	ROM 2	EGRESS TO MFCH		
119	0164	0304	1 100 010 011	PARA	BRN	MSTX2	PARA. REG. PRGM UX2, UX2	
120	0165	0277	1 011 111 111		BRN	MSTX		UX, UX, UX2
121	0166		0 000 001 100		PT0			
122	0167	0161	0 111 000 111		BRN	MSTORE	STORE UX IN R10	
123	0170	0000	0 000 000 011		BRN	MCAL10	170 UXUX2, "	
124	0171	0311	1 100 100 111		BRN	MSTXX2	UXX2, UXX2, UXUX2	
125	0172		0 001 001 100		PT1			
126	0173	0161	0 111 000 111		BRN	MSTORE	STORE UXX2 IN R11	
127	0174	0001	0 000 000 111		BRN	MCAL11	(UXX2)2, "	
128	0175	0003	0 000 001 111		BRN	MCAL12	UXUX2 – (UXX2)2 = DEN (C)	
129	0176		0 010 001 100		PT2			
130	0177	0161	0 111 000 111		BRN	MSTORE	STORE DEN (C) IN R12	
131	0200	0322	1 101 001 011		BRN	MSTX2Y	200 UX2Y, UX2Y	
132	0201		0 000 001 100		PT0			
133	0202	0163	0 111 001 111		BRN	MFCH	UX, UX2Y	
134	0203	0001	0 000 000 111		BRN	MCAL11	203 UXUX2Y, "	
135	0204	0326	1 101 011 011		BRN	MSTXY	UXY, UXY, UXUX2Y	
136	0205		0 001 001 100		PT1			
137	0206	0163	0 111 001 111		BRN	MFCH	UXX2, UXY, UXUX2Y	
138	0207	0001	0 000 000 111		BRN	MCAL11	UXX2UXY, ", UXUX2Y	
139	0210	0003	0 000 001 111		BRN	MCAL12	210 UXUX2Y – UXX2UXY, " = NUM(C)	
140	0211		0 010 001 100		PT2			
141	0212	0163	0 111 001 111		BRN	MFCH	DEN (C), NUM(C)	
142	0213	0007	0 000 011 111		BRN	MCAL17	C, C	
143	0214		0 010 001 100		PT2			
144	0215	0161	0 111 000 111		BRN	MSTORE	STORE C IN R12	
145	0216		0 001 001 100		PT1			
146	0217	0163	0 111 001 111		BRN	MFCH	UXX2, C	
147	0220	0001	0 000 000 111		BRN	MCAL11	220 CUXX2, CUXX2	
148	0221	0326	1 101 011 011		BRN	MSTXY	UXY, UXY, CUXX2	
149	0222	0003	0 000 001 111		BRN	MCAL12	CUXX2 – UXT, "	
150	0223		0 000 001 100		PT0			
151	0224	0163	0 111 001 111		BRN	MFCH	UX, CUXX2 – UXY	
152	0225	0366	1 111 011 011		BRN	MCAL18	UXY – CUXX2/UX = B	
153	0226		0 001 001 100		PT1			
154	0227	0161	0 111 000 111		BRN	MSTORE	STOER B IN R11	
155	0230	0253	1 010 101 111		BRN	MCALA	230 A CALC, AND STORED IN R10	
156	0231		0 000 000 000		DUMMY			
157	0232	0326	1 101 011 011		BRN	MSTXY	UXY, UXY	
158	0233		0 001 001 100		PT1			
159	0234	0163	0 111 001 111		BRN	MFCH	B,	
160	0235	0001	0 000 000 111		BRN	MCAL11	B UXY, "	
161	0236	0322	1 101 001 011		BRN	MSTX2Y	UX2Y, UX2Y, BUXY	
162	0237		0 010 001 100		PT2			
163	0240	0163	0 111 001 111		BRN	MFCH	240 C	
164	0241	0001	0 000 000 111		BRN	MCAL11	CUX2Y, ", BUXY	
165	0242	0011	0 000 100 111		BRN	MCAL19	CUX2Y+BUXY = NUM(R2)	
166	0243	0316	1 100 111 011		BRN	MSTY	UY, UY, NUM (R2)	
167	0244	0006	0 000 011 011		BRN	MCAL15	R2	
168	0245		0 011 001 100		PT3			
169	0246	0161	0 111 000 111		BRN	MSTORE	STORE R2 IN R13	
170	0247	0066	0 011 011 011		BRN	BPRINTS	LOAD STARTING ADDRESS FOR PRINTS PRGM	
171	0250		0 000 000 000		DUMMY		250	
172	0251		0 000 000 000		DUMMY			

STATISTICS FUNCTION BLOCK LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0252		0 000 000 000		DUMMY	
174	0253		0 010 001 100	MCALA	PT2	
175	0254	0357	1 111 000 001		JSB	CH0A SX2
176	0255		0 100 101 000		CTS	SX2, SX2
177	0256		0 010 001 100		PT2	
178	0257	0360	1 111 000 101		JSB	CH1A 257 C, SX2
179	0260	0355	1 110 111 001		JSB	MULT 260 CSX2, CSX2
180	0261		0 100 010 000	LINJ	ROM 2	EGRESS TO LINJ
181	0262		0 000 000 000		DUMMY	
182	0263		1 101 111 010	V2?	XS,AM1A	
183	0264		1 101 111 010		XS,AM1A	
184	0265		1 101 111 010		XS,AM1A	
185	0266	0373	1 111 101 111		BRN	FIN
186	0267		0 011 100 100		RS3	RETURN FROM ROM 7 ROUTINES
187	0270		0 000 110 000		RETURN	270
188	0271		1 100 101 000	SUITE1	DNR	
189	0272	0375	1 111 110 111		BRN	CONT
190	0273		0 000 000 000		DUMMY	
191	0274		0 000 000 000		DUMMY	
192	0275	0326	1 101 011 011		BRN	MSTXY ACCESS TO MSTXY
193	0276	0316	1 100 111 011		BRN	MSTY ACCESS TO MSTY
194	0277		0 010 001 100	MSTX	PT2	ACCESS TO MSTX CODE UX
195	0300		0 010 011 000		LDC2	300
196	0301		0 001 011 000		LDC1	
197	0302		0 001 011 000		LDC1	
198	0303	0332	1 101 101 011		BRN	LOAD
199	0304		0 010 001 100	MSTX2	PT2	CODE UX2
200	0305		0 111 011 000		LDC7	
201	0306		0 010 011 000		LDC2	
202	0307		0 010 011 000		LDC2	
203	0310	0332	1 101 101 011		BRN	LOAD 310
204	0311		0 010 001 100	MSTXX2	PT2	CODE UXX2
205	0312		0 110 011 000		LDC6	
206	0313		0 001 011 000		LDC1	
207	0314		0 010 011 000		LDC2	
208	0315	0332	1 101 101 011		BRN	LOAD
209	0316		0 010 001 100	MSTY	PT2	CODE UY
210	0317		0 101 011 000		LDC5	
211	0320		0 100 011 000		LDC4	320
212	0321	0331	1 101 100 111		BRN	LD4
213	0322		0 010 001 100	MSTX2Y	PT2	CODE X2Y
214	0323		1 000 011 000		LDC8	
215	0324		0 010 011 000		LDC2	
216	0325	0331	1 101 100 111		BRN	LD4
217	0326		0 010 001 100	MSTXY	PT2	CODE XY
218	0327		0 011 011 000		LDC3	
219	0330		0 001 011 000		LDC1	330
220	0331		0 100 011 000	LD4	LDC4	
221	0332		0 011 010 110	LOAD	MS,ZTC	
222	0333		1 010 001 100		PT10	333
223	0334		0 001 011 000		LDC1	LOAD END TEST DIGIT
224	0335		1 110 101 110		W,AXC	CODE WORD TO A
225	0336		1 111 001 100	NEUF	PT15	
226	0337		0 000 111 100	INC	PLS	
227	0340		1 101 111 010		XS,AM1A	340
228	0341	0337	1 101 111 111		BRN	INC DECODE XS TO SET POINTER
229	0342	0357	1 111 000 001		JSB	CH0A FETCH CODED REGISTER
230	0343		0 100 001 110		W,SLA	SHIFT TO NEXT CODED DIGIT
231	0344		1 101 111 110		S,AM1A	HAVE ALL 3 REGISTERS BEEN FETCHED?
232	0345	0106	0 100 011 011		BRN	STAT YES, CALC CORRECTED SUM OF SQUARES

STATISTICS FUNCTION BLOCK LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 100 101 000		CTS		NO, FETCH NEXT REGISTER
234	0347	0336	1 101 111 011		BRN	NEUF	
235	0350		0 001 001 100	MPRINTC	PT1		350
236	0351		0 001 011 000		LDC1		
237	0352	0100	0 100 000 011		BRN	PC	
238	0353		0 100 010 000	BOUT	ROM 2		
239	0354		0 011 000 100	DIVIDE	SS3		
240	0355		1 110 010 000		ROM 7		
241	0356		0 011 000 100	MULT	SS3		
242	0357		1 110 010 000		ROM 7		
243	0360		1 000 000 100	CH0A	SS8		360
244	0361		0 011 000 100	CH1A	SS3		
245	0362		1 110 010 000		ROM 7		
246	0363		0 011 111 110	MINUS	S,ZNCC		
247	0364		0 011 000 100	PLUS	SS3		
248	0365		1 110 010 000		ROM 7		
249	0366		0 011 111 110	MCAL18	S,ZNCC		
250	0367	0006	0 000 011 101		JSB	MCAL17	
251	0370		1 000 000 100	PRINT	SS8		370
252	0371		0 011 000 100		SS3		
253	0372		1 110 010 000		ROM 7		
254	0373		0 000 010 000	FIN	ROM 0		
255	0374		0 000 010 000	NOTE	ROM 0		
256	0375		0 000 010 000	CONT	ROM 0		
257	0376		0 010 010 000	SKIP+	ROM 1		
258	0377		1 010 010 000	XPRINT	ROM 5		377

STATISTICS FUNCTION BLOCK LISTING (ROM 4).

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		1 010 010 000	XROOT	ROM5		0
4	0001		0 101 001 100	MPRINTD	PT5		
5	0002		1 011 011 000		LDC11		
6	0003	0264	1 011 010 011		BRN	PCOM	
7	0004		0 000 011 100	MSCALEX	PRS		SET POINTER CODE FOR X SCALING
8	0005		0 000 011 100	MSCALEY	PRS		SET POINTER CODE FOR Y SCALING
9	0006		0 000 011 100	MDTIC	PRS		SET POINTER CODE FOR TICS
10	0007		0 000 011 100	MDAXES	PRS		SET POINTER CODE FOR AXES
11	0010		0 000 011 100		PRS		
12	0011		0 000 011 100		PRS		
13	0012		1 100 101 000		DNR		
14	0013		1 100 101 000		DNR		
15	0014		1 010 010 000	XPLOT	ROM 5		EGRESS TO XPLOT
16	0015		0 000 001 100	MLDNS	PT0		
17	0016	0357	1 111 000 001		JSB	CH0A	N1
18	0017		0 100 101 000		CTS		
19	0020		0 100 101 000		CTS		
20	0021		0 100 101 000		CTS		N1, N1, N1, N1
21	0022		1 001 111 010		XS,AM1		170 IS THIS IN 2 SAMPLE MODE?
22	0023	0375	1 111 110 111		BRN	CONT	NO
23	0024		1 001 001 100		PT9		YES
24	0025	0357	1 111 000 001		JSB	CH0A	N2
25	0026		1 100 101 000		DNR		N1, N1, N1, N2
26	0027	0163	0 111 001 111		BRN	SUITE1	
27	0030		0 010 101 000	VAR	CXM		170 VAR KEY
28	0031		1 011 001 100		PT11		4 to F11
29	0032		0 100 011 000		LDC4		
30	0033		0 010 101 000	EC	CXM		
31	0034	0373	1 111 101 111		BRN	FIN	
32	0035	0276	1 011 111 011	PT	BRN	MSTX	UX, UX
33	0036	0077	0 011 111 111		BRN	MP1?	SKIP 1 IF 2 VAR
34	0037	0101	0 100 000 111		BRN	MEXNS	EXCHANGE N1 AND N2
35	0040	0275	1 011 110 111		BRN	MSTY	UY, UY, UX
36	0041	0301	1 100 000 111		BRN	MCAL20	UX+UY,"
37	0042	0102	0 100 001 011		BRN	MP2?	SKIP 4 IF 2 VAR
38	0043	0101	0 100 000 111		BRN	MEXNS	EXCHANGE N1 AND N2
39	0044		1 100 101 000		DNR		BRING DOWN UX+UY FOR STORAGE
40	0045		0 100 001 100		PT4		
41	0046	0376	1 111 111 011		BRN	SKIP+	
42	0047	0274	1 011 110 011		BRN	MSTXY	UXY, UXY, UX+UY
43	0050	0277	1 011 111 111		BRN	MCAL30	UX+UY–2UXY,"
44	0051		0 101 001 100		PT5		120
45	0052	0161	0 111 000 111		BRN	MSTORE	STORE UX+UY (2 SAMP) OR UX+UY–2UXY (2 VAR.)
46	0053	0015	0 000 110 111		BRN	MLDNS	N1, N2, N2, N1 (2 SAMPLE CASE)
47	0054	0205	1 000 010 111		BRN	MCALNS	(N1+N2–2)/(UX+UY) (NH+N2) N1N2
48	0055	0000	0 000 000 011		BRN	XROOT	R( )
49	0056		0 101 001 100		PT5		
50	0057	0161	0 111 000 111		BRN	MSTORE	STORE ABOVE PARTIAL RESULT IN R15
51	0060	0015	0 000 110 111		BRN	MLDNS	127 N1, N2, N2, N1
52	0061	0227	1 001 011 111		BRN	MCALT	130 T

STATISTICS FUNCTION BLOCK LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062		0 101 001 100		PT5		
54	0063	0161	0 111 000 111		BRN	MSTORE	STORE T IN R15
55	0064	0162	0 111 001 011		BRN	MLDEQ	PRINT "/ ="
56	0065	0304	1 100 010 011		BRN	MPRINTT	
57	0066	0341	1 110 000 111		BRN	MDPRT	PRINT T
58	0067	0015	0 000 110 111		BRN	MLDNS	N1, N2, N2, N1
59	0070	0245	1 010 010 111		BRN	MCALD	DEGREES OF FREEDOM
60	0071		0 110 001 100		PT6		140
61	0072	0161	0 111 000 111		BRN	MSTORE	STORE DEG. FROM IN R16
62	0073	0162	0 111 001 011		BRN	MLDEQ	PRINT ": ="
63	0074	0001	0 000 000 111		BRN	MPRINTD	
64	0075	0341	1 110 000 111		BRN	MDPRT	PRINT DEGREES OF FEEDOM
65	0076	0353	1 110 101 111		BRN	BOUT	
66	0077		0 001 001 100	MP1?	PT1		
67	0100	0103	0 100 001 111		BRN	LT	
68	0101		0 010 010 000	MEXNS	ROM 1		EGRESS TO MEXNS
69	0102		0 100 001 100	MP2?	PT4		
70	0103		1 001 111 010	LT	XS,AM1		IS THIS 2 VAR OR 2 SAMPLE
71	0104	0376	1 111 111 011		BRN	SKIP+	2 VAR
72	0105	0375	1 111 110 111		BRN	CONT	2 SAMPLE
73	0106		1 101 100 010		P,AM1A		
74	0107	0111	0 100 100 111		BRN	R5	
75	0110		0 001 010 000		EERA		110
76	0111		1 010 010 000	R5	ROM 5		
77	0112		1 101 101 110	SEP	W,AM1A		20
78	0113		1 101 101 110		W,AM1A		IS THIS AN X RANGE ENTRY?
79	0114	0120	0 101 000 011		BRN	T2	NO
80	0115		0 111 011 000	BPRG1	LDC7		YES LOAD STARTING ADDRESS FOR PRG1
81	0116		0 101 011 000		LDC5		
82	0117	0155	0 110 110 111		BRN	ROMID	
83	0120		1 101 101 110	T2	W,AM1A		IS THIS A Y RANGE ENTRY?
84	0121	0125	0 101 010 111		BRN	T3	NO
85	0122		0 101 011 000	BPRG2	LDC5		30 YES, LOAD STARTING ADDRESS FOR PRG2
86	0123		1 110 011 000		LDC14		
87	0124	0155	0 110 110 111		BRN	ROMID	
88	0125		1 101 101 110	T3	W,AM1A		40 IS THIS A TIC ENTRY?
89	0126	0132	0 101 101 011		BRN	T4	NO
90	0127		1 101 011 000	BPTIC	LDC13		YES, LOAD STARTING ADDRESS FOR PTIC
91	0130		0 000 011 000		LDC0		
92	0131	0155	0 110 110 111		BRN	ROMID	
93	0132		1 101 101 110	T4	W,AM1A		IS THIS AN AXES ENTRY?
94	0133	0200	1 000 000 011		BRN	ZEROF11	NO, SET F11 FOR DATA ENTRY
95	0134		1 100 011 000	BPORG	LDC12		YES, LOAD STARTING ADDRESS FOR PORG
96	0135		1 010 011 000		LDC10		
97	0136	0155	0 110 110 111		BRN	ROMID	
98	0137	0005	0 000 010 111	RPG2	BRN	MSCALEY	STORE Y LIMITS
99	0140	0162	0 111 001 011		BRN	MLDEQ	
100	0141	0201	1 000 000 111		BRN	MPRINTG2	PRINT"RG2="
101	0142	0366	1 111 011 011		BRN	MSPRT	PRINT YMIN
102	0143	0341	1 110 000 111		BRN	MDPRT	PRINT YMAX
103	0144	0347	1 110 011 111		BRN	DATRES	SET F11 TO 0
104	0145		0 011 001 100	MPRINTI	PT3		
105	0146		1 011 011 000		LDC11		
106	0147		0 001 001 100		PT1		
107	0150	0331	1 101 100 111		BRN	LD6	
108	0151		1 100 101 110	AX	W,AXB		ACCESS TO AX FROM ROM 6, DIGIT TO A0
109	0152		0 001 001 100		PT1		SET POINTER TO LOAD STARTING ADDRESSES
110	0153	0112	0 100 101 011		BRN	SEP	
111	0154		1 100 011 000	BPT	LDC12		
112	0155		0 010 101 000	ROMID	CXM		

(164)

STATISTICS FUNCTION BLOCK LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 111 001 100		PT7		
114	0157		0 011 011 000		LDC3		
115	0160		0 000 010 000		ROM 0		160 EGRESS TO START
116	0161		0 100 010 000	MSTORE	ROM 2		
117	0162		0 100 010 000	MLDEQ	ROM 2		
118	0163		0 110 101 000	SUITE1	STA		N1, N1, N2, N2
119	0164		1 100 101 000		DNR		N1, N2, N2, N1
120	0165	0375	1 111 110 111		BRN	CONT	
121	0166	0004	0 000 010 011	PRG1	BRN	MSCALEX	STORE X LIMITS
122	0167	0162	0 111 001 011		BRN	MLDEQ	
123	0170	0262	1 011 001 011		BRN	MPRINTG1	PRINT "RG1="
124	0171	0366	1 111 011 011		BRN	MSPRT	PRINT XMIN
125	0172	0341	1 110 000 111		BRN	MDPRT	PRINT XMAX
126	0173	0347	1 110 011 111		BRN	DATRES	SET F11 to 0
127	0174		1 101 111 010	T	XS,AM1A		T KEY
128	0175		1 101 111 010		XS,AM1A		IS THIS IN 1 VARIABLE MODE?
129	0176	0333	1 101 101 111		BRN	COMP?	NO
130	0177	0373	1 111 101 111		BRN	FIN	YES, DO NOT ACCEPT
131	0200		1 100 010 000	ZEROF11	ROM 6		
132	0201	0265	1 011 011 001	MPRINTG2	JSB	RG	
133	0202		0 010 011 000		LDC2		
134	0203	0264	1 011 010 011		BRN	PCOM	150
135	0204		0 000 000 000		DUMMY		
136	0205	0355	1 110 111 001	MCALNS	JSB	MULT	N1N2, ", N2, N1
137	0206		1 100 101 000		DNR		
138	0207		1 100 101 000		DNR		N2, N1, N1N2, N1N2
139	0210	0363	1 111 010 001		JSB	PLUS	210 N1+N2, ", N1N2, N1N2
140	0211		0 100 101 000		CTS		N1+N2, N1+N2, N1+N2, N1N2
141	0212		0 011 001 110		W,ZTC		
142	0213		1 100 001 100		PT12		
143	0214		0 010 011 000		LDC2		2
144	0215	0362	1 111 001 101		JSB	MINUS	N1+N2–2, ", N1+N2, N1N2
145	0216		1 100 101 000		DNR		
146	0217		1 100 101 000		DNR		N1+N2, N1N2, N1+N2–2, N1+N2–2
147	0220	0355	1 110 111 001		JSB	MULT	220 N1N2(N1+N2) ",
148	0221		1 100 101 000		DNR		
149	0222	0353	1 110 110 001		JSB	DIVIDE	
150	0223		0 101 001 100		PT5		
151	0224	0360	1 111 000 101		JSB	CH1A	UX+UY
152	0225	0353	1 110 110 001		JSB	DIVIDE	(N1+N2–2)/(UX+UY) (NHN2)N1N2
153	0226	0375	1 111 110 111		BRN	CONT	
154	0227		0 001 001 100	MCALT	PT1		N1, N2, N2, N1
155	0230	0357	1 111 000 001		JSB	CH0A	230 SX
156	0231	0355	1 110 111 001		JSB	MULT	N2SX, "
157	0232		1 100 101 000		DNR		
158	0233		1 100 101 000		DNR		N2, N1, N2SX, "
159	0234		0 100 001 100		PT4		
160	0235	0357	1 111 000 001		JSB	CH0A	SY
161	0236	0355	1 110 111 001		JSB	MULT	N1SY, N1SY, N2SX, N2SX
162	0237		1 100 101 000		DNR		
163	0240	0362	1 111 001 101		JSB	MINUS	240 N2SX–N1SY, "
164	0241		0 101 001 100		PT5		
165	0242	0360	1 111 000 101		JSB	CH1A	
166	0243	0355	1 110 111 001		JSB	MULT	T=(N2SX–N1SY)R((N1+N2–2/(UX+UY) (N1+N2)N1N2)
167	0244	0375	1 111 110 111		BRN	CONT	
168	0245	0363	1 111 010 001	MCALD	JSB	PLUS	N1+N2, "
169	0246		0 011 001 110		W,ZTC		
170	0247		0 010 011 000		LDC2		250 2
171	0250	0362	1 111 001 101		JSB	MINUS	N1+N2–2," =DEG. FRDM.
172	0251		1 010 101 000		MTC		

STATISTICS FUNCTION BLOCK LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 001 111 010		XS,CM1		IS THIS IN 2 SAMPLE MODE?
174	0253	0256	1 010 111 011		BRN	HALF	NO, DIVIDE BY 2
175	0254		1 100 101 000		DNR		YES
176	0255	0375	1 111 110 111		BRN	CONT	
177	0256		0 011 001 110	HALF	W,ZTC		
178	0257		0 010 011 000		LDC2		257 2, 2N1–2
179	0260	0353	1 110 110 001		JSB	DIVIDE	260 N1–1 =DEG. FRDM
180	0261	0375	1 111 110 111		BRN	CONT	
181	0262	0265	1 011 011 001	MPRINTG1	JSB	RG	
182	0263		0 001 011 000		LDC1		
183	0264	0367	1 111 100 001	PCOM	JSB	PRINT	
184	0265	0375	1 111 110 111		BRN	CONT	
185	0266		0 101 001 100	RG	PT5		LOAD "RG" IN C
186	0267		1 100 011 000		LDC12		
187	0270		1 100 011 000		LDC12		270
188	0271		0 011 001 100		PT3		
189	0272		0 100 100 100		RS4		RETURN FROM ROM 7 ROUTINES
190	0273		0 000 110 000		RETURN		
191	0274		0 110 010 000	MSTXY	ROM 3		
192	0275		0 110 010 000	MSTY	ROM 3		
193	0276		0 110 010 000	MSTX	ROM 3		
194	0277	0363	1 111 010 001	MCAL30	JSB	PLUS	2UXY, ", UX+UY
195	0300		0 011 111 110		S,ZNCC		300 –2UXY, 2UXY, UX+UY
196	0301		0 110 101 000	MCAL20	STA		–2UXY; UX+UY
197	0302	0363	1 111 010 001		JSB	PLUS	UX+UY–2UXY
198	0303	0375	1 111 110 111		BRN	CONT	
199	0304		1 010 001 100	MPRINTT	PT10		
200	0305		1 010 011 000		LDC10		
201	0306		1 001 111 010		XS,AM1		
202	0307	0311	1 100 100 111		BRN	P	
203	0310	0264	1 011 010 011		BRN	PCOM	
204	0311		1 010 011 000	P	LDC10		
205	0312	0264	1 011 010 011		BRN	PCOM	
206	0313	0007	0 000 011 111	PORG	BRN	MDAXES	PLOT AXES
207	0314	0162	0 111 001 011		BRN	MLDEQ	
208	0315	0327	1 101 011 111		BRN	MPRINTO	PRINT "ORG="
209	0316	0366	1 111 011 011		BRN	MSPRT	320 PRINT X COOR. OF ORIGIN
210	0317	0341	1 110 000 111		BRN	MDPRT	PRINT Y COOR. OF ORIGIN
211	0320	0347	1 110 011 111		BRN	DATRES	SET F11 TO 0
212	0321	0006	0 000 011 011	PTIC	BRN	MDTIC	DRAW TICS
213	0322	0162	0 111 001 011		BRN	MLDEQ	PRINT "T I ="
214	0323	0145	0 110 010 111		BRN	MPRINTI	
215	0324	0366	1 111 011 011		BRN	MSPRT	PRINT X TIC INTERVAL
216	0325	0341	1 110 000 111		BRN	MDPRT	PRINT Y TIC INTERVAL
217	0326	0347	1 110 011 111		BRN	DATRES	SET F11 TO 0
218	0327		0 010 001 100	MPRINTO	PT2		
219	0330		1 011 011 000		LDC11		
220	0331		0 110 011 000	LD6	LDC6		
221	0332	0264	1 011 010 011		BRN	PCOM	
222	0333		0 011 001 100	COMP?	PT3		100
223	0334		1 001 100 010		P,AM1		IS THE DATA COMPLETE?
224	0335	0374	1 111 110 011		BRN	NOTE	NO, NOTE 30
225	0336		0 001 001 100		PT1		YES, LOAD STARTING ADDRESS FOR PT PRGM
226	0337		0 001 011 000		LDC1		
227	0340	0154	0 110 110 011		BRN	BPT	
228	0341		1 100 101 000	MDPRT	DNR		
229	0342	0377	1 111 111 111		BRN	XPRINT	
230	0343		0 100 000 100	CHP	SS4		
231	0344		1 110 010 000		ROM 7		
232	0345		0 100 000 100	DS	SS4		

(166)

STATISTICS FUNCTION BLOCK LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		1 110 010 000		ROM 7		
234	0347		0 010 101 000	DATRES	CXM	SET F11 TO 0	
235	0350		1 011 001 100		PT11		
236	0351		0 000 011 000		LDC0		
237	0352		0 010 101 000		CXM		
238	0353		0 100 010 000	BOUT	ROM 2		
239	0354		0 100 000 100	DIVIDE	SS4		
240	0355		1 110 010 000		ROM 7		
241	0356		0 100 000 100	MULT	SS4		
242	0357		1 110 010 000		ROM 7		
243	0360		1 000 000 100	CH0A	SS8		360
244	0361		0 100 000 100	CH1A	SS4		
245	0362		1 110 010 000		ROM 7		
246	0363		0 011 111 110	MINUS	S,ZNCC		
247	0364		0 100 000 100	PLUS	SS4		
248	0365		1 110 010 000		ROM 7		
249	0366	0344	1 110 010 101	MSPRT	JSB	DS	PRINT CONTENTS OF DS
250	0367	0377	1 111 111 111		BRN	XPRINT	
251	0370		1 000 000 100	PRINT	SS8		370
252	0371		0 100 000 100		SS4		
253	0372		1 110 010 000		ROM 7		
254	0373		0 000 010 000	FIN	ROM 0		
255	0374		0 000 010 000	NOTE	ROM 0		
256	0375		0 000 010 000	CONT	ROM 0		
257	0376		0 010 010 000	SKIP+	ROM 1		
258	0377		1 010 010 000	XPRINT	ROM 5		377

STATISTICS FUNCTION BLOCK LISTING (ROM 5)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 000 010 000	XPRINT	ROM 0		0
4	0001		0 000 010 000	XROOT	ROM 0		
5	0002		1 101 111 110	HIST	S,AM1A		
6	0003		1 101 111 110		S,AM1A		
7	0004		1 101 111 110		S,AM1A		
8	0005		1 101 111 110		S,AM1A		
9	0006	0373	1 111 101 111		BRN	FIN	
10	0007	0055	0 010 110 111		BRN	HIST1	
11	0010	0035	0 001 110 111		BRN	PLOT	10 PLOT KEY
12	0011		0 000 000 000		DUMMY		
13	0012	0365	1 111 011 001		JSB	BLANK	HIST KEY
14	0013	0002	0 000 001 011		BRN	HIST	
15	0014		1 011 001 100	XPENUP	PT11		SET POINTER CODE FOR PENUP
16	0015		0 000 101 100	XPLOT	YP0		20 ACCESS TO XPLOT
17	0016	0020	0 001 000 011		BRN	DECOP	
18	0017	0025	0 001 010 111		BRN	ADA	
19	0020		1 011 101 110	DECOP	W,ZTA		SET A0 TO PREVIOUSLY SET POINTER VALUE
20	0021		0 000 011 100	INC	PRS		
21	0022		1 111 101 110		W,AP1A		
22	0023		0 000 101 100		YP0		
23	0024	0021	0 001 000 111		BRN	INC	
24	0025		0 100 001 110	ADA	W,SLA		30 SHIFT A0 TO A1
25	0026		1 111 101 110		W,AP1A		LOAD U'S IN A0 AND A2
26	0027		1 111 111 010		XS,AP1A		
27	0030	0346	1 110 011 101		JSB	EXM712	EXCHANGE FLAGS
28	0031		0 010 101 000		CXM		
29	0032		0 010 011 000		LDC2		SET HENRY,S F11 TO 2 TO INDICATE I/O CALL
30	0033		0 010 101 000		CXM		
31	0034		0 000 100 000		RMGRA		EGRESS TO PLOT ROUTINES
32	0035		1 001 100 010	PLOT	P,AM1		IS THIS IN PLOTTING MODE
33	0036	0041	0 010 000 111		BRN	TO	YES
34	0037		0 111 001 100		PT7		NO, PRINT NOTE 37
35	0040	0374	1 111 110 011		BRN	NOTE	40
36	0041		1 101 111 110	TO	S,AM1A		HAS A REG. OR HIST. BEEN PERFORMED
37	0042	0045	0 010 010 111		BRN	T1	YES
38	0043		0 110 001 100		PT6		NO, PRINT NOTE 36
39	0044	0374	1 111 110 011		BRN	NOTE	
40	0045		1 101 001 100	T1	PT13		
41	0046		0 010 011 000		LDC2		
42	0047		0 001 001 100		PT1		
43	0050		0 001 011 110		S,AMC		50 WHAT KIND OF PLOT?
44	0051	0153	0 110 101 111		BRN	BPLOTH	HISTOGRAM
45	0052		0 111 011 000	BPLOT	LDC7		REG, LOAD STARTING ADDRESS FOR PLOR PRGM
46	0053		0 001 011 000		LDC1		53
47	0054	0155	0 110 110 111		BRN	ROMID	
48	0055	0365	1 111 011 001	HIST1	JSB	BLANK	PRINT HG
49	0056		0 001 001 100		PT1		
50	0057		0 101 011 000		LDC5		
51	0060		1 000 011 000		LDC8		
52	0061	0367	1 111 100 001		JSB	PRINT	

STATISTICS FUNCTION BLOCK LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0344	1 110 010 101		JSB	DS	0 TO DS
54	0063		0 011 001 110		W,ZTC		
55	0064		1 011 110 000		DTDS		
56	0065		0 001 001 100	BPHOUT	PT1		LOAD STARTING ADDRESS FOR PHOUT PRGM
57	0066		0 100 011 000		LDC4		
58	0067		0 000 011 000		LDC0		
59	0070	0155	0 110 110 111		BRN	ROMID	
60	0071	0167	0 111 011 111		BRN	MLB	CALC LOWER BOUND
61	0072	0000	0 000 000 011		BRN	XPRINT	PRINT LOWER BOUND
62	0073	0117	0 100 111 111		BRN	MFREN	FETCH FREQ
63	0074	0000	0 000 000 011		BRN	XPRINT	PRINT FREQ.
64	0075	0265	1 001 010 111		BRN	MRF	CALC REL FREQ
65	0076	0000	0 000 000 011		BRN	XPRINT	PRINT RELATIVE FREQ
66	0077	0223	1 001 001 111		BRN	MINCAD	ADV PAPER, INC I
67	0100	0325	1 101 010 111		BRN	MTI	IF 10 CELLS PRINTED, GO TO FIN
68	0101	0224	1 001 010 011	PHOUT	BRN	MINCI	INC I
69	0102	0000	0 000 000 011		BRN	XPRINT	100 PRINT CELL NUMBER
70	0103	0220	1 001 000 011		BRN	MDECI	
71	0104		1 100 001 100		PT12		
72	0105	0377	1 111 111 111		BRN	SKIP–	GO TO MLB
73	0106	0353	1 110 101 111		BRN	BOUT	
74	0107		1 100 101 000	MDPRT	DNR		XDATA TO C
75	0110		1 011 110 000		DTDS		110 STORE X DATA INDS
76	0111	0000	0 000 000 011		BRN	XPRINT	PRINT X DATA
77	0112		0 001 010 000		EERA		GO TO NEXT PROGRAM STEP
78	0113		0 000 000 000		DUMMY		
79	0114		0 000 000 000		DUMMY		
80	0115		0 000 000 000		DUMMY		
81	0116		0 000 000 000		DUMMY		
82	0117		0 010 010 000	MFREN	ROM 1		EGRESS TO MFREN
83	0120	0014	0 000 110 011	PLOTH	BRN	XPENUP	120 RAISE PEN
84	0121	0236	1 001 111 011		BRN	MPRE	0 TO DS
85	0122	0215	1 000 110 111		BRN	MEC	FREQ TO C
86	0123	0147	0 110 011 111		BRN	MLBO	0 TO C PEN TO COOR (LB, 0)
87	0124	0325	1 101 010 111		BRN	MTI	IF 10 CELLS PLOTTED GO TO FIN
88	0125	0200	1 000 000 011		BRN	MTLH	IF I EXCEEDS XMAX-1 ,GO TO FIN
89	0126	0150	0 110 100 011		BRN	XCOOR	130 PEN TO COOR. (LB, NEWF)
90	0127	0224	1 001 010 011		BRN	MINCI	
91	0130		0 111 001 100		PT7		
92	0131	0377	1 111 111 111		BRN	SKIP–	BACK TO MEC
93	0132	0014	0 000 110 011		BRN	XPENUP	
94	0133	0373	1 111 101 111		BRN	FIN	
95	0134		1 010 001 100	XPLINC	PT10		SET POINTER CODE FOR PEN TO COOR. WITH INCREMENT
96	0135	0020	0 001 000 011		BRN	DECOP	
97	0136	0107	0 100 011 111	PEV	BRN	MDPRT	PRINT X DATA AND STORE IT IN DS
98	0137		0 011 001 100		PT3		
99	0140	0161	0 111 000 111		BRN	MSTORE	140 STORE SDATA IN R13
100	0141	0243	1 010 001 111		BRN	MEVAL	Y
101	0142		0 100 001 100		PT4		
102	0143	0161	0 111 000 111		BRN	MSTORE	STORE Y IN R14
103	0144	0000	0 000 000 011		BRN	XPRINT	PRINT Y
104	0145	0353	1 110 101 111		BRN	BOUT	
105	0146	0341	1 110 000 111	EK	BRN	EVAL	EVAL KEY
106	0147		0 011 001 110	MLBO	W,ZTC		
107	0150		1 110 001 100	XCOOR	PT14		SET CODE FOR PLOT WITHOUT INCREMENT
108	0151	0020	0 001 000 011		BRN	DECOP	
109	0152		0 000 000 000		DUMMY		
110	0153		0 100 011 000	BPLOTH	LDC4		LOAD STARTING ADDRESS FOR PLOTH PRGM
111	0154		1 111 011 000		LDC15		
112	0155		0 010 101 000	ROMID	CXM		

STATISTICS FUNCTION BLOCK LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		0 111 001 100		PT7		
114	0157		0 100 011 000		LDC4		
115	0160		0 000 010 000		ROM 0		160
116	0161		0 100 010 000	MSTORE	ROM 2		
117	0162	0014	0 000 110 011	PLOR	BRN	XPENUP	RAISE PEN
118	0163	0277	1 011 111 111		BRN	MINIT	INITIALIZE REGRESSION PLOT
119	0164	0243	1 010 001 111		BRN	MEVAL	EVAL Y FOR CURRENT X IN DS
120	0165	0134	0 101 110 011		BRN	XPLINC	PEN TO (X,Y)
121	0166	0314	1 100 110 011		BRN	MTL	IF X IS GREATER THAN XMAX PLOT IS FINISH
122	0167	0344	1 110 010 101	MLB	JSB	DS	I,
123	0170		0 100 101 000		CTS		I,I
124	0171		0 101 001 100		PT5		
125	0172	0360	1 111 000 101		JSB	CH1A	CW,I
126	0173	0355	1 110 111 001		JSB	MULT	I(CW),",
127	0174		0 110 001 100		PT6		
128	0175	0360	1 111 000 101		JSB	CH1A	OFST ,I(CW)
129	0176	0363	1 111 010 001		JSB	PLUS	LB, LB,
130	0177	0375	1 111 110 111		BRN	CONT	
131	0200		0 010 001 100	MTLH	PT2		FETCH XMAX
132	0201	0342	1 110 001 101		JSB	CHP	
133	0202		0 100 101 000		CTS		
134	0203		0 011 001 110		W,ZTC		
135	0204		1 100 001 100		PT12		
136	0205		0 111 100 010		P,CP1C		1 ,XMAX
137	0206	0362	1 111 001 101		JSB	MINUS	XMAX–1
138	0207	0344	1 110 010 101		JSB	DS	1 ,XMAX–1
139	0210	0362	1 111 001 101		JSB	MINUS	XMAX–1–DS
140	0211		0 100 001 100		PT4		
141	0212		0 001 111 110		S,CM1		IS THERE ROOM TO PLOT ANOTHER WINDOW
142	0213	0376	1 111 111 011		BRN	SKIP+	NO ,PLOT IS FINISHED
143	0214	0117	0 100 111 111		BRN	MFREN	YES, CALC FREQ
144	0215		1 100 101 000	MEC	DNR		
145	0216		1 100 101 000		DNR		
146	0217	0150	0 110 100 011		BRN	XCOOR	
147	0220		1 011 101 110	MDECI	W,ZTA		
148	0221		1 101 111 110		S,AM1A		LOAD MINUS SIGN
149	0222	0224	1 001 010 101		JSB	CINC	
150	0223	0370	1 111 100 101	MINCA	DSB	ADV	
151	0224		1 011 101 110	MINCI	W,ZTA		
152	0225		1 100 001 100	CINC	PT12		
153	0226		1 111 100 010		P,AP1A		
154	0227		0 100 101 000		CTS		
155	0230		1 110 101 110		W,AXC		1 ,F
156	0231		0 100 101 000		CTS		1,1,F
157	0232	0344	1 110 010 101		JSB	DS	1 ,1 ,F
158	0233	0363	1 111 010 001		JSB	PLUS	I,I,F
159	0234		1 011 110 000		DTDS		
160	0235	0375	1 111 110 111		BRN	CONT	
161	0236	0344	1 110 010 101	MPRE	JSB	DS	
162	0237		0 011 001 110		W,ZTC		
163	0240		1 011 110 000		DTDS		0 TO DS
164	0241		0 001 001 100		PT1		
165	0242	0376	1 111 111 011		BRN	SKIP+	
166	0243	0344	1 110 010 101	MEVAL	JSB	DS	FETCH X
167	0244		0 100 101 000		CTS		X ,X
168	0245	0355	1 110 111 001		JSB	MULT	X2, X2
169	0246		0 010 001 100		PT2		
170	0247	0360	1 111 000 101		JSB	CH1A	C
171	0250	0355	1 110 111 001		JSB	MULT	CX2, CX2
172	0251	0344	1 110 010 101		JSB	DS	250 X

STATISTICS FUNCTION BLOCK LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 100 101 000		CTS		X, X, CX2,
174	0253		0 001 001 100		PT1		
175	0254	0360	1 111 000 101		JSB	CH1A	B
176	0255	0355	1 110 111 001		JSB	MULT	BX,BX,CX2
177	0256		0 110 101 000		STA		BX,CX2
178	0257	0363	1 111 010 001		JSB	PLUS	BX+CX2,"
179	0260		0 000 001 100		PT0		257
180	0261	0360	1 111 000 101		JSB	CH1A	260 A
181	0262	0363	1 111 010 001		JSB	PLUS	Y,Y
182	0263	0375	1 111 110 111		BRN	CONT	
183	0264		0 000 000 000		DUMMY		
184	0265		0 100 101 000	MRF	CTS		F,F
185	0266		0 000 001 100		PT0		
186	0267	0357	1 111 000 001		JSB	CH0A	N,F
187	0270	0353	1 110 110 001		JSB	DIVIDE	RF, RF
188	0271		0 111 101 010		X,CP1C		CONVERT TO %
189	0272		0 111 101 010		X,CP1C		
190	0273		0 000 000 000		DUMMY		270
191	0274	0375	1 111 110 111		BRN	CONT	
192	0275		0 101 100 100		RS5		RETURN FROM ROM 7 ROUTINES
193	0276		0 000 110 000		RETURN		
194	0277		0 011 001 110	MINIT	W,ZTC		LOAD 333.3
195	0300		1 100 001 100		PT12		
196	0301		0 011 011 000	LDC3	LDC3		
197	0302		1 000 101 100		YP8		
198	0303	0301	1 100 000 111		BRN	LDC3	
199	0304		0 000 001 100		PT0		
200	0305		0 010 011 000		LDC2		
201	0306		0 100 101 000		CTS		
202	0307		0 111 001 100		PT7		
203	0310	0342	1 110 001 101		JSB	CHP	FETCH X'
204	0311	0353	1 110 110 001		JSB	DIVIDE	GENERATE XINC
205	0312		1 101 001 100		PT13		
206	0313	0020	0 001 000 011		BRN	DECOP	
207	0314		0 111 001 100	MTL	PT7		FETCH HENRY'S FLAGS
208	0315	0360	1 111 000 101		JSB	CH1A	
209	0316		0 110 001 100		PT6		
210	0317		0 101 100 010		P,CM1C		HAS PLOT REACHED LIMIT
211	0320	0323	1 101 001 111		BRN	RF	YES
212	0321		0 011 001 100		PT3		NO
213	0322	0377	1 111 111 111		BRN	SKIP–	
214	0323		1 011 110 000	RF	DTDS		RESTORE HENRY'S FLAGS
215	0324	0373	1 111 101 111		BRN	FIN	
216	0325	0344	1 110 010 101	MTI	JSB	DS	ADDRESS DS
217	0326		0 100 101 000		CTS		
218	0327		0 011 001 110		W,ZTC		
219	0330		1 100 001 100		PT12		
220	0331		0 111 101 110		W,CP1C		
221	0332		0 111 100 010		P,CP1C		10,1
222	0333	0362	1 111 001 101		JSB	MINUS	I–10,"
223	0334		0 101 001 100		PT5		
224	0335		0 110 111 110		S,ZMC		IS I LESS THAN 10?
225	0336	0376	1 111 111 011		BRN	SKIP+	NO, PRGM IS FINISHED
226	0337	0375	1 111 110 111		BRN	CONT	
227	0340	0155	0 110 110 111		BRN	ROMID	340 ACCESS TO ROM 5 ROMID
228	0341		1 010 000 100	EVAL	SS10		EGRESS TO ROM 7 EVAL ROUTINE
229	0342		1 110 010 000		ROM 7		
230	0343		0 101 000 100	CHP	SS5		
231	0344		1 110 010 000		ROM 7		
232	0345		0 101 000 100	DS	SS5		

STATISTICS FUNCTION BLOCK LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		1 110 010 000		ROM 7		
234	0347		0 101 000 100	EXM712	SS5		
235	0350		1 110 010 000		ROM 7		350
236	0351		0 000 000 000		DUMMY		
237	0352		0 000 000 000		DUMMY		
238	0353		0 100 010 000	BOUT	ROM 2		
239	0354		0 101 000 100	DIVIDE	SS5		
240	0355		1 110 010 000		ROM 7		
241	0356		0 101 000 100	MULT	SS5		
242	0357		1 110 010 000		ROM 7		
243	0360		1 000 000 100	CH0A	SS8		360
244	0361		0 101 000 100	CH1A		SS5	
245	0362		1 110 010 000		ROM 7		
246	0363		0 011 111 110	MINUS	S,ZNCC		
247	0364		0 101 000 100	PLUS	SS5		
248	0365		1 110 010 000		ROM 7		
249	0366		0 101 000 100	BLANK	SS5		
250	0367		1 110 010 000		ROM 7		
251	0370		1 000 000 100	PRINT	SS8		370
252	0371		0 101 000 100	ADV	SS5		
253	0372		1 110 010 000		ROM 7		
254	0373		0 000 010 000	FIN	ROM 0		
255	0374		0 000 010 000	NOTE	ROM 0		
256	0375		0 000 010 000	CONT	ROM 0		
257	0376		0 010 010 000	SKIP+	ROM 1		
258	0377		0 000 010 000	SKIP–	ROM 0		377

(172)

STATISTICS FUNCTION BLOCK LISTING (ROM 6)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 010 101 000		CXM		0 1 TO F5 TO INDICATE DELETE
4	0001		0 101 001 100		PT5		
5	0002		0 001 011 000		LDC1		
6	0003		0 010 101 000		CXM		
7	0004	0365	1 111 011 001		JSB	BLANK	PRINT "DEL"
8	0005		1 101 001 100		PT13		
9	0006		1 011 011 000		LDC11		
10	0007		1 011 001 100		PT11		
11	0010		1 001 011 000		LDC9		10
12	0011	0365	1 111 011 001		JSB	BLANK	
13	0012		0 001 001 100		PT1		
14	0013		0 010 011 000		LDC2		
15	0014	0367	1 111 100 001		JSB	PRINT	
16	0015	0373	1 111 101 111		BRN	FIN	
17	0016	0342	1 110 001 101	AXES	JSB	CHP	AXES KEY, FETCH PLOTTER REG2
18	0017		0 100 101 000		CTS		SAVE REG2
19	0020		0 001 011 000		LDC1		20 1 TO C6
20	0021		1 011 110 000		DTDS		STORE IN REG2
21	0022	0253	1 010 101 111		BRN	SUITE1	
22	0023		1 101 001 100	RECALL	PT13		RECALL KEY, LOAD 10 IN C13
23	0024		1 010 011 000		LDC10		
24	0025	0302	1 100 001 011		BRN	SUITE2	
25	0026		1 001 111 010	SAMPLE	XS,AM1		SAMPLE KEY IS THIS IN 2 SAMPLE MODE?
26	0027	0373	1 111 101 111		BRN	FIN	NO
27	0030		0 011 001 110		W,ZTC		30 YES
28	0031		0 100 001 100		PT4		
29	0032		1 001 100 010		P,AM1		WHICH IS THE NEW SAMPLE MODE?
30	0033	0323	1 101 001 111		BRN	D1	DATA 1 1 TO A0
31	0034		0 111 101 110		W,CP1C		DATA 2 2 TO A0
32	0035	0323	1 101 001 111		BRN	D1	
33	0036		0 011 001 100	DELETE	PT3		DELETE KEY
34	0037	0375	1 111 110 111		BRN	DEL	
35	0040		0 110 001 100		PT6		40 DATA ENTRY KEY
36	0041		1 101 100 010		P,AM1A		IS THIS VARIABLE DATA?
37	0042	0147	0 110 011 111		BRN	T1	NO
38	0043		1 101 111 010	VAREN	XS,AM1A		YES, IS THIS IN 2 SAMPLE MODE?
39	0044	0051	0 010 100 111		BRN	TV1	NO
40	0045		0 100 001 100		PT4		YES
41	0046		1 001 100 010		P,AM1		
42	0047	0151	0 110 100 111		BRN	BPYACC	
43	0050		0 010 010 000	BPXACC	ROM 1		50 EGRESS TO BPXACC
44	0051		1 001 111 010	TV1	XS,AM1		IS THIS IN 1 VARIABLE MODE?
45	0052	0216	1 000 111 011		BRN	TOGGLE	NO
46	0053		1 101 111 110	HACC	S,AM1A		53 YES, IS THIS IN HISTOGRAM MODE?
47	0054	0056	0 010 111 011		BRN	HG	YES
48	0055	0050	0 010 100 011		BRN	BPXACC	NO
49	0056		1 100 101 000	HG	DNR		DATA
50	0057		0 100 101 000		CTS		
51	0060		0 100 101 000		CTS		60 DATA, DATA, DATA
52	0061		0 110 001 100		PT6		

STATISTICS FUNCTION BLOCK LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0360	1 111 000 101		JSB	CH1A	OFFSET, DATA, DATA
54	0063	0362	1 111 001 101		JSB	MINUS	DATA-OFFSET, ", DATA
55	0064		0 101 001 100		PT5		
56	0065	0360	1 111 000 101		JSB	CH1A	CW
57	0066	0353	1 110 110 001		JSB	DIVIDE	DATA-OFFSET/CW=I
58	0067		0 100 101 000		CTS		I, I, I, DATA
59	0070		0 001 111 110		S,CM1		70 IS I NEG?
60	0071	0110	0 100 100 011		BRN	PRINTL	YES, PRINT "L"
61	0072		0 011 001 110		W,ZTC		NO
62	0073		1 100 001 100		PT12		
63	0074		0 111 101 110		W,CP1C		
64	0075		0 111 100 010		P,CP1C		10, I, I, DATA
65	0076	0362	1 111 001 101		JSB	MINUS	I—10, I—10,I, DATA
66	0077		0 001 111 110		S,CM1		IS I LESS THAN 10
67	0100	0115	0 100 110 111		BRN	DAAD	100 YES, DATA IS IN RANGE OF HISTOGRAM
68	0101	0365	1 111 011 001		JSB	BLANK	NO, PRINT "H"
69	0102	0365	1 111 011 001		JSB	BLANK	
70	0103	0105	0 100 010 111		BRN	SUITE5	
71	0104		0 000 010 000	OUT	ROM 0		
72	0105		0 001 001 100	SUITE5	PT1		EGRESS TO OUT
73	0106		0 101 011 000		LDC5		
74	0107	0267	1 011 011 111		BRN	CP	
75	0110	0365	1 111 011 001	PRINTL	JSB	BLANK	110
76	0111	0365	1 111 011 001		JSB	BLANK	
77	0112		0 000 001 100		PT0		
78	0113		0 111 011 000		LDC7		
79	0114	0267	1 011 011 111		BRN	CP	
80	0115		1 100 101 000	DAAD	DNR		
81	0116		1 100 101 000		DNR		I, DATA
82	0117		1 001 000 100		SS9		SET FLAG FOR SHALK ROUTINE
83	0120		0 010 010 000		ROM 1		120 EGRESS TO CHALK
84	0121		1 011 001 100	DIGIT	PT11		
85	0122		1 101 000 010		P,AMCA		IS A11 LESS THAN 5
86	0123	0227	1 001 011 111		BRN	CHAR	NO
87	0124		1 100 101 110	VARDEC	W,AXB		YES, A11=4, VAR. NUM. ENTRY, DIGIT TO A0
88	0125		0 100 001 110		W,SLA		DIGIT TO A2
89	0126		0 100 001 110		W,SLA		
90	0127		0 010 001 100		PT2		127
91	0130		0 011 011 000		LDC3		130 3 TO C2
92	0131		0 010 001 100		PT2		
93	0132		0 001 011 010		XS,AMC		IS VARIABLE NUMBER GREATER THAN 2?
94	0133	0374	1 111 110 011		BRN	NOTE	YES, NOTE 32
95	0134	0370	1 111 100 101		JSB	ADV	NO
96	0135		0 011 001 110		W,ZTC		0 TO STAT REGISTERS 00 THROUGH 17
97	0136		0 111 101 110		W,CP1C		BUILD ADDRESS FOR R17
98	0137		1 100 001 100	PT12			
99	0140		1 011 011 000		LDC11		140
100	0141		0 110 011 000		LDC6		
101	0142		1 011 001 100		PT11		
102	0143	0344	1 110 010 101		JSB	LOOP	INITIALIZE REGISTERS 10 TO 17
103	0144		1 100 001 100		PT12		BUILD ADDRESS FOR R09
104	0145		1 010 011 000		LDC10		
105	0146	0152	0 110 101 011		BRN	SUITE3	
106	0147		0 100 010 000	TI	ROM 2		EGRESS TO OFFSET/WIDTH ENTRIES
107	0150		1 000 010 000	AX	ROM 4		150 EGRESS TO AX
108	0151		0 010 010 000	BPYACC	ROM 1		
109	0152	0344	1 110 010 101	SUITE3	JSB	LOOP	INITIALIZE REGISTERS 00 TO 09
110	0153		1 110 101 110		W,AXC		INITIALIZED FLAGS TO M, ALL FLAGS 0
111	0154		0 110 001 110		W,CTA		EXCEPT F2 WHICH MAY BE 0, 1 OR 2
112	0155		0 010 101 000		CXM		

STATISTICS FUNCTION BLOCK LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156	0365	1 111 011 001		JSB	BLANK	PRINT VARIABLE ENTRY MODE
114	0157	0365	1 111 011 001		JSB	BLANK	
115	0160		1 000 001 100		PT8		160 LOAD DEC. POINTS IN C
116	0161		1 010 011 000	L2	LDC10		
117	0162		0 001 101 100		YP1		
118	0163	0161	0 111 000 111		BRN	L2	
119	0164		1 101 111 010		XS,AM1A		IS THIS IN 2 SAMPLE MODE?
120	0165	0206	1 000 011 011		BRN	ONETWO	NO
121	0166		1 011 001 100		PT11		YES, PRINT "2T"
122	0167		0 010 011 000		LDC2		
123	0170		1 010 011 000		LDC10		170
124	0171	0370	1 111 100 101		JSB	ADV	
125	0172	0367	1 111 100 001		JSB	PRINT	
126	0173	0365	1 111 011 001		JSB	BLANK	PRINT "DATA 1"
127	0174	0365	1 111 011 001		JSB	BLANK	
128	0175	0333	1 101 110 001		JSB	DATA	
129	0176		0 001 011 000	VARP1	LDC1		
130	0177	0370	1 111 100 101	VARP	JSB	ADV	
131	0200	0367	1 111 100 001		JSB	PRINT	200
132	0201		0 010 101 000	ZEROF11	CXM		SET F11 FOR VARIABLE DATA ENTRY
133	0202		1 011 001 100		PT11		
134	0203		0 011 000 010		P,ZTC		203
135	0204		0 010 101 000	FIM	CXM		
136	0205	0373	1 111 101 111		BRN	FIN	
137	0206		1 101 001 100	ONETWO	PT13		PRINT V1 OR V2
138	0207		0 101 011 000		LDC5		LOAD "V2" IN C
139	0210		0 000 000 000		DUMMY		210
140	0211		0 010 011 000		LDC2		
141	0212		1 100 001 100		PT12		
142	0213		1 001 111 010		XS,AM1		IS THIS IN 1 VARIABLE MODE?
143	0214	0177	0 111 111 111		BRN	VARP	NO PRINT "V2"
144	0215	0176	0 111 111 011		BRN	VARP1	YES PRINT "V1"
145	0216		0 010 101 000	TOGGLE	CXM		SET CALCULATION FLAG F13 TO 0
146	0217		0 011 011 110		S,ZTC		AND TOGGLE COMPLETENESS FLAG F3
147	0220		0 011 001 100		PT3		220
148	0221		0 011 100 010		P,ZNCC		
149	0222		0 010 101 000		CXM		
150	0223		1 001 100 010		P,AM1		WHICH VARIABLE IS BEING ENTERED?
151	0224	0151	0 110 100 111		BRN	BPYACC	Y, GO TO PRGM FOR Y ACCUMULATION
152	0225	0370	1 111 100 101		JSB	ADV	X
153	0226	0050	0 010 100 011		BRN	BPXACC	GO TO PRGM FOR X ACCUMULATION
154	0227		1 101 100 010	CHAR	P,AM1A		IS THIS A CHARACTER DIGIT ENTRY
155	0230	0150	0 110 100 011		BRN	AX	230 NO
156	0231		1 100 101 110		W,AXB		YES DIGIT TO A0
157	0232		0 000 001 100		PT0		DIGIT TO F0
158	0233		1 010 101 000		MTC		
159	0234		1 110 100 010		P,AXC		
160	0235		0 110 000 010		P,CTA		
161	0236		0 010 101 000		CXM		
162	0237	0365	1 111 011 001		JSB	BLANK	PRINT "CT" A0
163	0240		0 000 001 100		PT0		240
164	0241		0 100 001 110	SHIFT	W,SLA		
165	0242		0 000 111 100		PLS		
166	0243		1 101 101 100		YP13		
167	0244	0241	1 010 000 111		BRN	SHIFT	
168	0245		1 110 100 010		P,AXC		
169	0246	0365	1 111 011 001		JSB	BLANK	
170	0247		0 001 001 100		PT1		
171	0250		0 001 011 000		LDC1		250
172	0251		1 010 011 000		LDC10		

STATISTICS FUNCTION BLOCK LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
173	0252	0177	0	111	111	111		BRN	VARP
174	0253		0	111	001	100	SUITE1	PT7	
175	0254		1	011	111	000		DSTC	RETREIVE FROM REG 2
176	0255		0	110	101	110		W,ZMC	IS C0?
177	0256	0374	1	111	110	011		BRN	NOTE YES, PLOTTER WAS NOT PRESENT NOTE 37
178	0257		1	100	101	000		DNR	257 NO, REG2 DATA TO C
179	0260		1	011	110	000		DTDS	260 RETURN ORIGINAL DATA
180	0261		0	010	101	000		CXM	6 TO F11 TO INDICATE AXES DIGIT ENTRY
181	0262		1	011	001	100		PT11	
182	0263		0	110	011	000		LDC6	
183	0264		1	000	001	100		PT8	1 TO F8 TO INDICATE PLOTTING MODE
184	0265		0	001	011	000		LDC1	
185	0266	0204	1	000	010	011		BRN	FIM
186	0267	0367	1	111	100	001	CP	JSB	PRINT
187	0270		1	100	101	000		DNR	270
188	0271		1	100	101	000		DNR	
189	0272	0050	0	010	100	011		BRN	BPXACC I, DATA
190	0273		0	011	001	110	SUITE4	W,ZTC	
191	0274		1	011	110	000		DTDS	0 TO ADDRESSED REGISTER
192	0275		1	100	101	000		DNR	BRING DOWN ADDRESS
193	0276		0	101	100	010		P,CM1C	DECREMENT ADDRESS; WAS ADDRESS 0?
194	0277	0345	1	110	010	111		BRN	LOOP NO, CONTINUE TO INITIALIZE THIS CHIP
195	0300		0	110	100	100		RS6	300 RETURN FROM ROM 7 ROUTINES
196	0301		0	000	110	000		RETURN	
197	0302		0	001	101	010	SUITE2	X,CM1	IS THE EXPONENT 0
198	0303	0313	1	100	101	111		BRN	L11 NO
199	0304		1	001	001	110		W,SRC	YES, SHIFT REGHT, MANTISSA OF ADDRESS IS READY
200	0305		0	011	001	010	AD	X,ZTC	SET EXPONENT TO 1
201	0306		0	111	101	010		X,CP1C	
202	0307		1	001	110	000		ATDS	
203	0310		0	000	110	100		CLS	310
204	0311		1	011	111	000		DSTC	CONTENTS OF SELECTED REGISTER TO C
205	0312	0104	0	100	010	011		BRN	OUT
206	0313		1	011	011	000	L11	LDC11	LOAD 11 IN C12, MANTISSA OF ADDRESS IS READY
207	0314	0305	1	100	010	111		BRN	AD
208	0315		0	000	000	000		DUMMY	ADDRESS SELECTED REGISTER
209	0316		0	000	000	000		DUMMY	RETURN D-DATA TO D
210	0317		0	000	000	000		DUMMY	REGISTER DATA TO C
211	0320		0	000	000	000		DUMMY	320
212	0321		0	000	000	000		DUMMY	
213	0322		0	000	000	000		DUMMY	
214	0323		0	111	101	110	D1	W,CP1C	
215	0324		1	110	101	110		W,AXC	
216	0325		0	011	100	010		P,ZNCC	
217	0326		0	010	101	000		CXM	
218	0327	0365	1	111	011	001		JSB	BLANK PRINT "DATA" A0
219	0330	0365	1	111	011	001		JSB	BLANK 330
220	0331	0333	1	101	110	001		JSB	DATA
221	0332		1	110	100	010		P,AXC	
222	0333	0176	0	111	111	101		JSB	VARP
223	0334		0	101	001	100	DATA	PT5	LOAD WORD DATA IN C
224	0335		1	011	011	000		LDC11	
225	0336		1	011	011	000		LDC11	
226	0337		1	011	011	000		LDC11	
227	0340		1	011	011	000		LDC11	340
228	0341		0	000	001	100		PT0	
229	0342		0	000	110	000		RETURN	
230	0343		0	110	000	100	CHP	SS6	
231	0344		1	110	010	000		ROM 7	
232	0345		1	001	110	000	LOOP	ATDS	ADDRESS REGISTER

STATISTICS FUNCTION BLOCK LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 100 101 000		CTS		ADDRESS TO D
234	0347	0273	1 011 101 111		BRN	SUITE4	
235	0350		1 011 001 100	DE	PT11		350 DIGIT INTERPRETATION
236	0351		0 101 011 000		LDC5		5 TO C11
237	0352	0121	0 101 000 111		BRN	DIGIT	
238	0353		0 000 000 000		DUMMY		
239	0354		0 110 000 100	DIVIDE	SS6		
240	0355		1 110 010 000		ROM 7		
241	0356		0 000 000 000		DUMMY		
242	0357		0 000 000 000		DUMMY		
243	0360		0 000 000 000		DUMMY		360
244	0361		0 110 000 100	CH1A	SS6		
245	0362		1 110 010 000		ROM 7		
246	0363		0 011 111 110	MINUS	S,ZNCC		
247	0364		0 110 000 100	PLUS	SS6		
248	0365		1 110 010 000		ROM 7		
249	0366		0 110 000 100	BLANK	SS6		
250	0367		1 110 010 000		ROM 7		
251	0370		1 000 000 100	PRINT	SS8		370
252	0371		0 110 000 100	ADV	SS6		
253	0372		1 110 010 000		ROM 7		
254	0373		0 000 010 000	FIN	ROM 0		
255	0374		0 000 010 000	NOTE	ROM 0		
256	0375		1 001 100 010	DEL	P,AM1		IS DATA SET COMPLETE?
257	0376	0374	1 111 110 011		BRN	NOTE	NO, PRINT NOTE33
258	0377	0370	1 111 100 101		JSB	ADV	377 YES

STATISTICS FUNCTION BLOCK LISTING (ROM 7)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
3	0000		0 101 110 000		PRE		NO PRINT (S8) OR ADVANCE	
4	0001		1 011 010 100	WAIT	YS11		WAIT FOR PRINT FLAG	
5	0002	0001	0 000 000 111		BRN	WAIT		
6	0003		1 011 100 100		RS11			
7	0004		1 000 010 100		YS8		IS THIS A PRINT OR AN ADVANCE?	
8	0005	0013	0 000 101 111		BRN	ADV	ADVANCE	
9	0006		1 101 110 000		TCS		PRINT, LOAD ID REG.	
10	0007		1 111 110 000		CCS		10	
11	0010		0 000 011 100		PRS			
12	0011		1 110 101 100		YP14		IS PRINT FINISHED?	
13	0012	0001	0 000 000 111		BRN	WAIT	NO, WAIT FOR NEXT FLAG	
14	0013		0 011 110 000	ADV	ADV		YES, ADVANCE PAPER	
15	0014	0254	1 010 110 011		BRN	TERM		
16	0015		0 010 101 000	STUP	CXM		FLAG BACK TO M — DATA BACK TO C	
17	0016	0254	1 010 110 011		BRN	TERM		
18	0017		1 101 011 000	BLK	LDC13			
19	0020		1 111 101 100		YP15		FILL C WITH BLANKS	
20	0021	0017	0 000 111 111		BRN	BLK	20	
21	0022	0254	1 010 110 011		BRN	TERM		
22	0023		0 111 111 110	INC10	S,CP1C		SET C13 TO POINTER VALUE	
23	0024		0 000 011 100		PRS			
24	0025		0 000 101 100	TO	YP0			
25	0026	0023	0 001 001 111		BRN	INC10		
26	0027		1 001 001 110		W,SRC		C13 TO C11	
27	0030		1 001 001 110		W,SRC			
28	0031		0 111 101 110		W,CP1C		30	
29	0032		1 100 001 100		PT12			
30	0033		1 000 010 100		YS8		WHICH CHIP SOUGHT?	
31	0034	0037	0 001 111 111		BRN	LD11	CHIP11	
32	0035		1 010 011 000		LDC10		CHIP 10	
33	0036	0040	0 010 000 011		BRN	AD		
34	0037		1 011 011 000	LD11	LDC11			
35	0040		1 001 110 000	AD	ATDS			
36	0041		1 100 001 100		PT12			
37	0042	0237	1 001 111 111		BRN	COMT		
38	0043		0 110 101 000	SUM1	STA			
39	0044		0 000 101 110		W,ZTB			
40	0045		1 111 111 010		XS,AP1A			
41	0046		1 111 111 010		XS,AP1A			
42	0047		0 111 111 010		XS,CP1C			
43	0050		0 111 111 010		XS,CP1C		50	
44	0051		0 001 001 010		X,AMC			
45	0052	0054	0 010 110 011		BRN	SUM2		
46	0053		1 110 101 110		W,AXC		53	
47	0054		1 110 100 110	SUM2	M,AXC			
48	0055		0 110 100 110		M,ZMC			
49	0056	0060	0 011 000 011		BRN	SUM3		
50	0057		1 110 101 110		W,AXC			
51	0060		1 000 100 110	SUM3	M,BXC		60	
52	0061		0 001 001 010	SUM4	X,AMC			

STATISTICS FUNCTION BLOCK LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0070	0 011 100 011		BRN	SUM5	
54	0063		1 010 001 110		W,SRB		
55	0064		1 111 101 010		X,AP1A		
56	0065		0 000 001 110		W,ZMB		
57	0066	0070	0 011 100 011		BRN	SUM5	
58	0067	0061	0 011 000 111		BRN	SUM4	
59	0070		0 101 111 010	SUM5	XS,CM1C		70
60	0071		0 101 111 010		XS,CM1C		
61	0072		1 011 101 010		X,ZTA		
62	0073		1 101 011 110		S,AMCA		
63	0074		1 001 111 110		S,AM1		
64	0075	0077	0 011 111 111		BRN	SUM6	
65	0076	0142	0 110 001 011		BRN	MPY26	
66	0077		1 000 000 110	SUM6	M,AMB		
67	0100	0103	0 100 001 111		BRN	SUM7	100
68	0101		0 011 111 110		S,ZNCC		
69	0102		1 100 101 110		W,AXB		
70	0103		1 100 001 110	SUM7	W,AMBA		
71	0104		0 000 000 000		DUMMY		
72	0105	0152	0 110 101 011		BRN	NRM21	
73	0106		0 110 101 000	LDV2	STA		
74	0107		0 000 101 110		W,ZTB		
75	0110		0 101 001 010		X,AMCC		110
76	0111		0 101 011 110		S,AMCC		
77	0112	0114	0 100 110 011		BRN	DIV22	
78	0113		0 010 111 110		S,ZMCC		
79	0114		1 100 100 110	DIV22	M,AXB		
80	0115		1 011 101 110		W,ZTA		
81	0116		1 100 101 100		YP12		
82	0117	0143	0 110 001 111		BRN	MPY27	
83	0120		0 001 100 110		M,CM1		120
84	0121	0124	0 101 010 011		BRN	DIV23	
85	0122		0 001 001 100	NOTE31	PT1		ZERO DIVISOR OR OVERFLOW
86	0123	0203	1 000 001 111		BRN	NOTE	
87	0124		1 000 110 010	DIV23	WP,BXC		
88	0125		1 110 100 110		M,AXC		
89	0126	0130	0 101 100 011		BRN	DIV15	
90	0127		0 111 100 010	DIV14	P,CP1C		127
91	0130		1 100 010 110	DIV15	MS,AMBA		130
92	0131	0127	0 101 011 111		BRN	DIV14	
93	0132		1 110 010 110		MS,APBA		
94	0133		0 100 010 110		MS,SLA		
95	0134		0 000 011 100	DIV16	PRS		
96	0135		0 000 101 100		YP0		
97	0136	0130	0 101 100 011		BRN	DIV15	
98	0137		0 110 001 110	TNM12	W,CTA		
99	0140		0 010 001 010		X,BTC		140
100	0141	0152	0 110 101 011		BRN	NRM21	
101	0142		1 110 001 110	MPY26	W,APBA		
102	0143		0 101 100 010	MPY27	P,CM1C		
103	0144	0142	0 110 001 011		BRN	MPY26	
104	0145		1 011 001 110	MPY28	W,SRA		
105	0146		0 000 111 100		PLS		
106	0147		1 101 101 100		YP13		
107	0150	0143	0 110 001 111		BRN	MPY27	150
108	0151		0 111 101 010		X,CP1C		
109	0152		1 011 111 110	NRM21	S,ZTA		
110	0153		1 100 001 100		PT12		
111	0154		0 000 101 110		W,ZTB		
112	0155		1 001 100 010	NRM23	P,AM1		

STATISTICS FUNCTION BLOCK LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156	0164	0 111 010 011		BRN	NRM24	
114	0157		0 100 001 110		W,SLA		
115	0160		0 101 101 010		X,CM1C		160
116	0161		1 001 101 110		W,AM1		
117	0162	0155	0 110 110 111		BRN	NRM23	
118	0163		0 011 001 110		W,ZTC		
119	0164		0 100 101 010	NRM24	X,ATB		
120	0165		1 110 001 110		W,APBA		
121	0166		1 001 111 110		S,AM1		
122	0167	0145	0 110 010 111		BRN	MPY28	
123	0170		1 110 100 110		M,AXC		170
124	0171		0 110 001 110	NRM25	W,CTA		
125	0172		0 000 101 110		W,ZTB		
126	0173		1 100 001 100		PT12		
127	0174		0 001 100 110	OFL1	M,CM1		
128	0175	0206	1 000 011 011		BRN	OFL2	
129	0176	0220	1 001 000 011		BRN	OFL4	
130	0177		0 011 001 110	DS1	W,ZTC		
131	0200		1 111 011 000		LDC15		200 BUILD ADDRESS OF
132	0201		1 001 011 000		LDC9		DEDICATED STORAGE IN C
133	0202	0234	1 001 110 011		BRN	AC	
134	0203		0 000 110 100	NOTE	CLS		203 INSURE RETURN TO ROM 0
135	0204		0 000 010 000		ROM 0		AFTER NOTE SUBROUTINES
136	0205		0 000 010 000	ZERO	ROM 0		RETURN TO ROM 0
137	0206		0 110 111 010	OFL2	XS,ZMC		
138	0207	0221	1 001 000 111		BRN	CLOSE	
139	0210		0 010 101 010		X,ZMCC		210
140	0211		0 101 111 010		XS,CM1C		
141	0212	0215	1 000 110 111		BRN	OFL3	
142	0213		1 110 101 110		W,AXC		
143	0214	0221	1 001 000 111		BRN	CLOSE	
144	0215		1 110 111 010	OFL3	XS,AXC		
145	0216		1 001 111 010		XS,AM1		
146	0217	0122	0 101 001 011		BRN	NOTE31	
147	0220		0 011 001 110	OFL4	W,ZTC		220
148	0221		0 100 101 000	CLOSE	CTS		
149	0222	0254	1 010 110 011		BRN	TERM	
150	0223		0 011 001 110	CHP1	W,ZTC		SET CB TO POINTER VALUE
151	0224		0 111 111 110	INC12	S,CP1C		
152	0225		0 000 011 100		PRS		
153	0226		0 000 101 100		YP0		
154	0227	0224	1 001 010 011		BRN	INC12	
155	0230		1 001 001 110		W,SRC		230
156	0231		1 001 001 110		W,SRC		CB TO C11
157	0232		1 100 001 100		PT12		
158	0233	0340	1 110 000 011		BRN	AC1	BUILD REMAINDER OF ADDRESS FOR PLOTTER DS
159	0234		0 111 101 010	AC	X,CP1C		
160	0235		1 001 110 000		ATDS		
161	0236		0 110 001 100		PT6		
162	0237		1 011 111 000	COMT	DSTC		
163	0240	0254	1 010 110 011		BRN	TERM	240
164	0241		1 011 011 000	EX	LDC11		
165	0242		0 111 011 000		LDC7		
166	0243		1 001 110 000		ATDS		
167	0244		0 111 001 100		PT7		
168	0245		1 011 111 000		DSTC		FETCH HENRY'S FLAG
169	0246		1 001 011 000		LDC9		250
170	0247		1 100 001 100		PT12		
171	0250		0 001 011 000		LDC1		SET TO PRGM MODE AND RMGRB USER
172	0251		0 010 101 000		CXM		HENRY'S FLAG TO M, STAT FLAG TO C

STATISTICS FUNCTION BLOCK LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		1 011 110 000		DTDS		STORE STAT FLAG
174	0253		1 100 101 000		DNR		DATA TO C
175	0254		1 000 100 100	TERM	RS8		
176	0255		1 011 100 100		RS11		
177	0256		0 001 010 100		YS1		FIND CALLING ROM AND
178	0257	0261	1 011 000 111		BRN	TEST2	257 RETURN CONTROL
179	0260		0 010 010 000		ROM1		260
180	0261		0 010 010 100	TEST2	YS2		
181	0262	0264	1 011 010 011		BRN	TEST3	
182	0263		0 100 010 000		ROM 2		
183	0264		0 011 010 100	TEST3	YS3		
184	0265	0267	1 011 011 111		BRN	TEST4	
185	0266		0 110 010 000		ROM 3		
186	0267		0 100 010 100	TEST4	YS4		
187	0270	0272	1 011 101 011		BRN	TEST5	270
188	0271		1 000 010 000		ROM 4		
189	0272		0 101 010 100	TEST5	YS5		
190	0273	0275	1 011 110 111		BRN	TEST6	
191	0274		1 010 010 000		ROM 5		
192	0275		0 110 010 100	TEST6	YS6		
193	0276	0300	1 100 000 011		BRN	TEST10	
194	0277		1 100 010 000		ROM 6		
195	0300		1 010 010 100	TEST10	YS10		300
196	0301	0205	1 000 010 111		BRN	ZERO	
197	0302		0 000 110 000		RETURN		
198	0303		1 101 111 110	EVAL1	S,AM1A		
199	0304		1 101 111 110		S,AM1A		IS THIS A LINEAR REGRESSION?
200	0305	0323	1 101 001 111		BRN	T2	NO
201	0306		1 101 001 100		PT13		YES, PRINT LE
202	0307		1 011 011 000		LDC11		
203	0310	0367	1 111 100 001		JSB	BLANK	310
204	0311		0 000 001 100		PT0		
205	0312		0 111 011 000		LDC7		
206	0313	0372	1 111 101 101	CP	JSB	P/A	ADVANCE PAPER
207	0314		1 000 000 100		SS8		
208	0315	0372	1 111 101 101		JSB	P/A	PRINT
209	0316	0346	1 110 011 101		JSB	DS	
210	0317		0 001 001 100	BPEV	PT1		LOAD STARTING ADDRESS FOR PEV PRGM.
211	0320		0 101 011 000		LDC5		320
212	0321		1 101 011 000		LDC13		
213	0322	0336	1 101 111 011		BRN	ROMID	
214	0323		0 110 001 100	T2	PT6		
215	0324		1 101 111 110		S,AM1A		IS THIS A PARABOLIC REGRESSION?
216	0325	0203	1 000 001 111		BRN	NOTE	NO, PRINT NO 36
217	0326	0367	1 111 100 001		JSB	BLANK	YES, PRINT PE
218	0327		1 001 001 100		PT9		
219	0330		1 010 011 000		LDC10		330
220	0331		1 100 011 000		LDC12		
221	0332	0313	1 100 101 111		BRN	CP	
222	0333		0 000 000 000		DUMMY		333
223	0334		0 000 000 000		DUMMY		
224	0335		0 000 000 000		DUMMY		
225	0336		1 010 100 100	ROMID	RS10		
226	0337		1 010 010 000		ROM 5		EGRESS TO ROMID IN ROM 5
227	0340		1 101 011 000	AC1	LDC13		
228	0341	0234	1 001 110 011		BRN	AC	
229	0342		0 000 000 000		DUMMY		
230	0343	0367	1 111 100 001	EVAL	JSB	BLANK	ACCESS TO EVAL
231	0344	0303	1 100 001 111		BRN	EVAL1	
232	0345		0 000 000 000	CHP	DUMMY		CHP ACCESS

STATISTICS FUNCTION BLOCK LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346	0223	1 001 001 111		BRN	CHP1	
234	0347		1 100 001 100	DS	PT12		DS ACCESS
235	0350	0177	0 111 111 111		BRN	DS1	350
236	0351		0 100 101 000	EXM712	CTS		EXM712 ACCESS
237	0352		0 011 001 110		W,ZTC		ADDRESS FLAG VECTOR
238	0353		0 111 101 110		W,CP1C		
239	0354		1 100 001 100		PT12		
240	0355	0241	1 010 000 111		BRN	EX	
241	0356		1 100 001 100	DIVIDE	PT12		DIVIDE ACCESS
242	0357	0106	0 100 011 011		BRN	LDV2	
243	0360		0 010 101 010	MULT	X,ZMCC		360 MULT ACCESS
244	0361		0 011 001 100		PT3		
245	0362	0106	0 100 011 011		BRN	LDV2	
246	0363		0 011 001 110	CH	W,ZTC		CH0A/CH1A ACCESS
247	0364		0 000 000 000		DUMMY		
248	0365	0025	0 001 010 111		BRN	TO	
249	0366		1 100 001 100	PLUS	PT12		PLUS ACCESS
250	0367	0043	0 010 001 111		BRN	SUM1	
251	0370		0 111 110 000	BLANK	CTT		370 BLANK ACCESS
252	0371		1 101 001 100		PT13		
253	0372	0017	0 000 111 111		BRN	BLK	
254	0373		0 010 101 000	P/A	CXM		PRINT/ADV ACCESS
255	0374		1 100 001 100		PT12		
256	0375		0 001 100 010		P,CM1		IS THIS IN PRGM MODE?
257	0376	0015	0 000 110 111		BRN	STUP	YES, DONT PRINT OR ADVANCE
258	0377		0 010 101 000		CXM		NO CONTINUE

TYPEWRITER INTERFACE LISTING (ROM 0)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		0 000 000 000		DUMMY				
4	0001		0 000 000 000		DUMMY				
5	0002		0 000 000 000		DUMMY				
6	0003		0 000 000 000		DUMMY				
7	0004		0 000 000 000		DUMMY				
8	0005		0 000 000 000		DUMMY				
9	0006		0 000 000 000		DUMMY				
10	0007		0 000 000 000		DUMMY				
11	0010		0 000 000 000		DUMMY				
12	0011		0 000 000 000		DUMMY				
13	0012		1 000 010 000	DIV	ROM 4			GO TO ROM 4	
14	0013	0313	1 100 101 111	MUL	BRN	MUL 1		CONTINUE ROUTINE	
15	0014		1 110 010 000	UP	ROM 7			GO TO ROM 7 TO TEST OP LEVEL	
16	0015		0 100 010 000	XMEM	ROM 2			GO TO ROM 2 FOR ERROR MESSAGE, NO MEMORY	
17	0016		0 100 010 000	CPAR	ROM 2			GO TO ROM 2 TO PROCESS CLOSED PAR.	
18	0017		0 011 010 000	ACKEY0	TKRA			ACCEPT KEYCODE	
19	0020		1 000 010 000	EQUAL	ROM 4			GO TO ROM 4	
20	0021		0 100 010 000	SYNTAX	ROM 2			GO TO ROM 2 TO PROCESS SYNTAX	
21	0022		0 100 010 000	DECPT	ROM 2			GO TO ROM 2 TO PROCESS DECPT	
22	0023		0 100 010 000	DIGIT0	ROM 2			GO TO ROM 2 TO PROCESS DIGIT	
23	0024		0 010 010 000	ACKEY1	ROM 1			GO TO ROM 1 TO ACCEPT KEYCODE	
24	0025	0344	1 110 010 011		BRN	STEP1		CONTINUE ROUTINE	
25	0026		0 100 010 000	OPAR	ROM 2			GO TO ROM 2 TO PROCESS OPEN PAR.	
26	0027		1 011 101 110	PMS5	W,ZTA			CLEAR A-REG	
27	0030		0 010 101 000		CXM			DATA TO C-REG, FLAG TO M-REG	
28	0031		0 001 010 000		EERA			ACCEPT STARTING ADDRESS	
29	0032		1 111 101 010	DIGIT3	X,AP1A			LOAD DIGIT 3 IN A-REG	
30	0033		1 111 101 010	DIGIT2	X,AP1A			LOAD DIGIT 2 IN A-REG	
31	0034		1 111 101 010	DIGIT1	X,AP1A			LOAD DIGIT 1 IN A-REG	
32	0035	0023	0 001 001 111		BRN	DIGIT0		GO TO DIGIT 0 ENTRY	
33	0036		1 010 000 100	PERCENT	SS10			SET EXCHANGE FLAG	
34	0037	0056	0 010 111 011		BRN	STORE		GO TO STORE ROUTINE	
35	0040	0317	1 100 111 111	PLUS	BRN	PLUS1		CONTINUE ROUTINE	
36	0041		0 110 010 000	FLGEXC	ROM 3			GO TO ROM 3 FOR SUBROUTINE	
37	0042		1 111 101 010	DIGIT6	X,AP1A			LOAD DIGIT 6 IN A-REG	
38	0043		1 111 101 010	DIGIT5	X,AP1A			LOAD DIGIT 5 IN A-REG	
39	0044		1 111 101 010	DIGIT4	X,AP1A			LOAD DIGIT 4 IN A-REG	
40	0045	0032	0 001 101 011		BRN	DIGIT3		GO TO DIGIT 3 ENTRY	
41	0046		1 000 101 110	RECALL	W,BXC			SAVE DATA IN B-REG	
42	0047	0315	1 100 110 111		BRN	RECALL1		CONTINUE ROUTINE	
43	0050	0277	1 011 111 111	MINUS	BRN	MINUS1		CONTINUE ROUTINE	
44	0051		0 000 000 000		DUMMY				
45	0052		1 111 101 010	DIGIT9	X,AP1A			LOAD DIGIT 9 IN A-REG	
46	0053		1 111 101 010	DIGIT8	X,AP1A			LOAD DIGIT 8 IN A-REG	
47	0054		1 111 101 010	DIGIT7	X,AP1A			LOAD DIGIT 7 IN A-REG	
48	0055	0042	0 010 001 011		BRN	DIGIT6		GO TO DIGIT 6 ENTRY	
49	0056		0 010 010 000	STORE	ROM 1			GO TO ROM 1 TO PROCESS STORE	
50	0057		0 000 000 000		DUMMY				
51	0060		0 000 000 000		DUMMY				
52	0061		0 000 000 000		DUMMY				

TYPEWRITER INTERFACE LISTING (ROM 0) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
53	0062	0172	0	111	101	011	RUN	BRN	RUN1	GO TO RUN ROUTINE
54	0063	0102	0	100	001	011	LAST	BRN	LAST1	GO TO LAST ENTRY ROUTINE
55	0064	0366	1	111	011	011	START	BRN	START1	CONTINUE ROUTINE
56	0065	0111	0	100	100	111		BRN	SUPVR2	RETURN TO SUPERVISOR FROM DIGEX
57	0066		1	000	101	110	CANCEL	W,BXC		SAVE DATA IN B-REG
58	0067	0162	0	111	001	011		BRN	CANCEL1	CONTINUE ROUTINE
59	0070	0111	0	100	100	111	SHIFT	BRN	SUPVR2	ACCEPT SHIFT KEY, NOP
60	0071		0	000	000	000		DUMMY		
61	0072	0141	0	110	000	111	RND	BRN	RND1	CONTINUE ROUTINE
62	0073	0321	1	101	000	111	RESET	BRN	RESET1	CONTINUE ROUTINE
63	0074		0	000	100	000	RESET2	RMGRA		GO TO ROM GROUP A RESET ROUTINE
64	0075		0	000	000	000		DUMMY		
65	0076		1	000	101	110	PRINT	W,BXC		SAVE DATA IN B-REG
66	0077		0	010	010	000		ROM 1		GO TOO ROM 1 PRINT ROUTINE
67	0100		0	000	000	000		DUMMY		
68	0101		0	000	110	000		RETURN		RETURN TO CALLING ROUTINE
69	0102	0277	1	100	000	001	LAST1	JSB	OPLEV=0?	DOES OPLEVEL = 0 ?
70	0103		0	111	000	100		SS7		SET FIRST DIGIT ENTRY FLAG
71	0104	0040	0	010	000	101		JSB	FLGEXC	CALL D. S. FLAG REG AND EXCHANGE FLAGS
72	0105		1	000	101	110		W,BXC		RESTORE DATA IN C-REG
73	0106		0	000	100	000		RMGRA		RETURN TO MAIN ROM GROUP
74	0107		0	011	001	110	ENTER1	W,ZTC		CLEAR C-REG
75	0110		1	000	101	110	SUPVR1	W,BXC		RESTORE DATA TO C-REG
76	0111		0	010	101	000	SUPVR2	CXM		PREPARE TO TEST OP MODE DIGIT
77	0112		1	100	001	100	SUPVR3	PT12		SELECT OP MODE DIGIT
78	0113		0	001	100	010	SUPVR4	P,CM1		DOES OP MODE = 0?
79	0114	0256	1	010	111	011		BRN	PMS1	NO, GO TO PROGRAM MODE SUPERVISOR
80	0115		0	010	101	000		CXM		YES, RESTORE DATA IN C-REG
81	0116		0	010	010	000	DISPLAY	ROM 1		GO TO ROM 1 FOR DISPLAY ROUTINE
82	0117		0	000	100	000	1C	RMGRA		CONTINUE POWER ON
83	0120		1	001	000	000		IS1		BEGIN POWER ON
84	0121	0200	1	000	000	011		BRN	IOPWO1	GO TO TURN-OFF ROUTINE
85	0122		0	000	100	000	2C	RMGRA		CONTINUE POWER ON
86	0123		1	001	000	000		IS1		BEGIN POWER ON
87	0124	0202	1	000	001	011		BRN	IOPWO2	GO TO TURN-ON ROUTINE
88	0125		0	000	100	000	3C	RMGRA		CONTINUE POWER ON
89	0126		1	001	000	000		IS1		BEGIN POWER ON
90	0127	0147	0	110	100	001		JSB	PRESET	PRESET FLAG
91	0130	0125	0	101	010	111		BRN	3C	CONTINUE POWER ON
92	0131	0363	1	111	001	111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
93	0132	0355	1	110	111	001	MMS1	JSB	KEYUP	WAIT FOR OLD KEY UP CONDITION
94	0133		0	101	000	000	MMS2	IS2		PREPARE TO TURN BUSY LIGHT OFF
95	0134		0	100	100	000		RBL		TURN BUSY LIGHT OFF
96	0135		1	001	000	000		IS1		WAIT FOR KEY DOWN
97	0136	0237	1	001	111	111		BRN	MMS6	CONTINUE SUPERVISOR
98	0137		0	000	000	000		DUMMY		
99	0140	0363	1	111	001	111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
100	0141		0	010	101	000	RND1	CXM		PREPARE TO CLEAR FLAGS
101	0142		1	000	001	100		PT8		SET POINTER TO FIRST FLAG
102	0143		0	000	011	000		LDC0		CLEAR CR/LF FLAG
103	0144		0	000	011	000		LDC0		CLEAR TAB FLAG
104	0145	0112	0	100	101	011		BRN	SUPVR3	RETURN TO SUPERVISOR
105	0146		0	000	000	000		DUMMY		
106	0147		0	000	000	000		DUMMY		
107	0150		1	011	010	100	PRESET	YS11		NORMAL CALL ?
108	0151	0153	0	110	101	111		BRN	PRESET1	NO, POWER ON
109	0152	0214	1	000	110	011		BRN	SCODE	YES, CONTINUE
110	0153		0	011	001	110	PRESET1	W,ZTC		CLEAR C-REG
111	0154		0	010	001	100		PT2		SET POINTER TO RND
112	0155		0	010	011	000		LDC2		SET RND TO 2

TYPEWRITER INTERFACE LISTING (ROM 0) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156		1 001 001 100		PT9		SET POINTER TO SELECT CODE
114	0157		0 010 011 000		LDC2		SET SELECT CODE TO 2
115	0160		0 010 101 000		CXM		STORE FLAG
116	0161	0041	0 010 000 111		BRN	FLGEXC	STORE FLAG IN DATA STORAGE
117	0162		0 011 001 110	CANCEL1	W,ZTC		CLEAR C-REG
118	0163		0 111 110 000		CTT		CLEAR T-REG
119	0164		1 100 101 000		DNR		CHECK STACK CONTENTS
120	0165		0 001 100 110		M,CM1		REAL DATA ?
121	0166	0170	0 111 100 011		BRN	CANCEL2	YES, RESTORE DATA
122	0167		0 011 001 110		W,ZTC		NO, CLEAR DATA ENTERED
123	0170		0 100 101 000	CANCEL2	CTS		RESTORE STACK
124	0171	0107	0 100 011 111		BRN	ENTER1	RETURN TO SUPERVISOR
125	0172		0 010 101 000	RUN1	CXM		FLAG TO C-REG FOR UPDATE
126	0173		1 100 001 100		PT12		SELECT OP MODE DIGIT
127	0174		0 001 011 000		LDC1		SET OP MODE TO 1
128	0175		1 010 100 000		SBL		SET BUSY LIGHT
129	0176	0355	1 110 111 001	RUN2	JSB	KEYUP	WAIT FOR KEYUP CONDITION BEFORE PROCEEDING
130	0177	0271	1 011 100 111		BRN	PMS2	GO TO INC. AND READ ROUTINE
131	0200	0147	0 110 100 001	IOPWO1	JSB	PRESET	PRESET FLAG
132	0201	0117	0 100 111 111		BRN	1C	CONTINUE POWER ON
133	0202	0147	0 110 100 001	IOPWO2	JSB	PRESET	PRESET FLAG
134	0203	0122	0 101 001 011		BRN	2C	CONTINUE POWER ON
135	0204		0 110 100 100	INITIAL	RS6		RESET STATUS BIT 6
136	0205		1 000 100 100		RS8		RESET STATUS BIT 8
137	0206		1 010 100 100		RS10		RESET STATUS BIT 10
138	0207		1 011 100 100		RS11		RESET STATUS BIT 11
139	0210		0 111 100 100	INITIAL1	RS7		RESET STATUS BIT 7
140	0211	0040	0 010 000 101	ENTER	JSB	FLGEXC	CALL D.S. FLAG REG AND EXCHANGE FLAGS
141	0212	0107	0 100 011 111		BRN	ENTER1	CONTINUE ROUTINE
142	0213	0162	0 111 001 011		BRN	CANCEL1	CONTINUE
143	0214		0 011 001 110	SCODE	W,ZTC		PREPARE TO CALL D.S.
144	0215		0 111 101 110		W,CP1C		BEGIN LOADING ADDRESS
145	0216		1 100 001 100		PT12		SET POINTER
146	0217		1 110 011 000		LDC14		LOAD ADDRESS
147	0220		1 001 110 000		ATDS		ENABLE REGISTER
148	0221		1 001 001 100		PT9		SET POINTER TO SELECT CODE DIGIT
149	0222		1 011 111 000		DSTC		RECALL FLAG
150	0223		1 001 001 110	SHIFTSC	W,SRC		SHIFT SELECT CODE TO PROPER POSITION
151	0224		0 000 011 100		PRS		SHIFT POINTER
152	0225		0 000 101 100		YP0		IS SHIFT DONE ?
153	0226	0223	1 001 001 111		BRN	SHIFTSC	NO, CONTINUE SHIFT
154	0227		0 101 000 010		P,AMCC		YES, PREPARE SELECT CODE TEST
155	0230		0 001 100 010		P;CM1		IS THIS THE RIGHT SELECT CODE ?
156	0231	0304	1 100 010 011		BRN	RET	NO, EXIST FROM THIS CHANNEL
157	0232		1 001 100 100		RS9		YES, RESET STATUS BIT 9
158	0233		0 100 100 100		RS4		RESET STATUS BIT 4
159	0234		0 101 100 100		RS5		RESET STATUS BIT 5
160	0235	0204	1 000 010 011		BRN	INITIAL	CONTINUE
161	0236		0 000 000 000		DUMMY		
162	02373		0 000 010 100	MMS6	YS0		IS NEW KEY DOWN?
163	0240	0237	1 001 111 111		BRN	MMS6	NO, WAIT FOR NEW KEY
164	0241		0 000 100 100		RS0		YS, WAIT FOR BOUNCE TO SETTLE
165	0242		0 000 001 100		PT0		BEGIN DELAY ROUTINE
166	0243		0 000 111 100	MMS4	PLS		COUNTDOWN LOOP
167	0244		0 011 101 100		YP3		IS COUNTDOWN COMPLETE ?
168	0245	0243	1 010 001 111		BRN	MMS4	NO, CONTINUE TO WAIT FOR 3 MSEC.
169	0246		1 100 100 000		YFKB		WAS IT FUNCTION BLK KEY ?
170	0247		1 000 101 000	MMS5	DSOF		TURN DISPLAY OFF
171	0250		0 000 101 110		W,ZTB		CLEAR B-REG
172	0251		1 011 101 110		W,ZTA		CLEAR A-REG

TYPEWRITER INTERFACE LISTING (ROM 0) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
173	0252		1 011 010 100		YS11		TEST FUNCTION BLOCK FLAG
174	0253	0017	0 000 111 111		BRN	ACKEY0	NO, MAIN UNIT KEY
175	0254		1 011 100 100		RS11		YES, RESET FLAG
176	0255	0024	0 001 010 011		BRN	ACKEY1	GO TO ROM 1 TO ACCEPT KEY
177	0256		0 101 100 100	PMS1	RS5		RESET STATUS BIT 5
178	0257		0 000 010 100		YS0		IS A KEY DOWN ? (STOP PROGRAM)
179	0260	0271	1 011 100 111		BRN	PMS2	NO, CONTINUE ROUTINE
180	0261		0 000 100 100		RS0		YES, RESET STATUS BIT
181	0262		0 010 101 000		CXM		RESTORE DATA
182	0263		0 010 101 000	PSTOP	CXM		BEGIN MAN. AND PROG. STOP
183	0264		1 100 001 100		PT12		SELECT OP MODE DIGIT
184	0265		0 011 000 010		P,ZTC		SET OP MODE TO 0 (MANUAL)
185	0266	0112	0 100 101 101		JSB	SUPVR4	GO TO SUPERVISOR
186	0267	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
187	0270	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
188	0271		0 111 000 000	PMS2	PINC		INCREMENT PROGRAM COUNTER
189	0272		1 000 100 000		READ		READ NEXT PROGRAM INSTRUCTION
190	0273		0 110 011 110		S,CTA		LOAD OP LEVEL IN A-REG FOR TEST
191	0274	0305	1 100 010 111		BRN	PMS4	CONTINUE ROUTINE
192	0275		0 000 000 000		DUMMY		
193	0276		0 010 010 000	POCTFLG	ROM 1		GO TO ROM 1
194	0277		0 100 010 000	MINUS1	ROM 2		GO TO ROM 2 TO CONTINUE HSPACEB
195	0300		1 000 101 110	OPLEV=0?	W,BXC		SAVE DATA IN B-REG
196	0301		1 010 101 000		MTC		RECALL TFLAG FOR OP LEVEL TEST
197	0302		0 001 111 110		S,CM1		DOES OP LEVEL=0?
198	0303	0021	0 001 000 111		BRN	SYNTAX	NO, GO TO SYNTAX ERROR ROUTINE
199	0304		0 000 110 000	RET	RETURN		YES, RETURN TO CALLING ROUTINE
200	0305		1 101 111 110	PMS4	S,AM1A		DOES OP LEVEL = 0?
201	0306		1 101 111 110		S,AM1A		DOES OP LEVEL = 1?
202	0307		1 101 111 110		S,AM1A		DOES OP LEVEL = 2?
203	0310	0320	1 101 000 011		BRN	PMS3	NO, CONTINUE TEST
204	0311	0276	1 011 111 011		BRN	POCTFLG	YES, GO AND PROCESS OCTAL DATA
205	0312		0 000 000 000		DUMMY		
206	0313		1 001 000 100	MUL1	SS9		SET STATUS BIT FOR FAST SPACE
207	0314	0341	1 110 000 111		BRN	MUL2	CONTINUE ROUTINE
208	0315		0 010 010 000	RECALL1	ROM 1		GO TO ROM 1 TO FINISH RECALL ROUTINE
209	0316		0 000 000 000		DUMMY		
210	0317		1 110 010 000	PLUS1	ROM 7		GO TO ROM 7 TO TEST OP LEVEL
211	0320		0 110 010 000	PMS3	ROM 3		GO TO ROM 3 TO CONTINUE TEST
212	0321		0 010 101 000	RESET1	CXM		RECALL TFLAG
213	0322		0 011 011 110		S,ZTC		CLEAR OP LEVEL
214	0323		0 010 101 000		CXM		STORE NEW OP LEVEL
215	0324	0040	0 010 000 101		JSB	FLGEXC	CALL D.S. FLAG REG AND EXCHANGE FLAGS
216	0325		0 100 000 100		SS4		SET STATUS BIT FOR RMGRA CLEAR ROUTINE
217	0326		1 011 001 100		PT11		SET POINTER FOR ROMGRA CLEAR ROUTINE
218	0327	0074	0 011 110 011		BRN	RESET2	CONTINUE ROUTINE
219	0330		0 000 000 000		DUMMY		
220	0331	0360	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
221	0332	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
222	0333	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
223	0334	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
224	0335	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
225	0336	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
226	0337	0363	1 111 001 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
227	0340		0 000 000 000		DUMMY		
228	0341		0 100 010 000	MUL2	ROM 2		GO TO ROM 2 TO CONTINUE HSPACEF
229	0342		0 000 000 000		DUMMY		
230	0343		0 000 000 000		DUMMY		
231	0344		0 010 101 000	STEP1	CXM		PREEPARE DATA FOR OP LEVEL TEST
232	0345	0176	0 111 111 011	STEP2	BRN	RUN2	WAIT FOR KEYUP

TYPEWRITER INTERFACE LISTING (ROM Ø) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	Ø346		Ø ØØØ ØØØ ØØØ		DUMMY		
234	Ø347		Ø ØØØ ØØØ ØØØ		DUMMY		
235	Ø35Ø		Ø ØØØ ØØØ ØØØ		DUMMY		
236	Ø351		Ø ØØØ ØØØ ØØØ		DUMMY		
237	Ø352		Ø ØØØ ØØØ ØØØ		DUMMY		
238	Ø353		Ø ØØØ ØØØ ØØØ		DUMMY		
239	Ø354		Ø ØØØ ØØØ ØØØ		DUMMY		
240	Ø355		Ø ØØØ ØØØ ØØØ		DUMMY		
241	Ø356		Ø ØØØ Ø1Ø 1ØØ	KEYUP	YSØ		IS KEY DOWN ?
242	Ø357	Ø3Ø4	1 1ØØ Ø1Ø Ø11		BRN	RET	NO, RETURN TO CALLING ROUTINE
243	Ø36Ø		Ø ØØØ 1ØØ 1ØØ		RSØ		YES, RESET STATUS BIT AND CHECK KEY DOWN AGAIN
244	Ø361	Ø356	1 11Ø 111 Ø11		BRN	KEYUP	CHECK KEY STATUS
245	Ø362	Ø363	1 111 ØØ1 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
246	Ø363		1 ØØØ 1Ø1 11Ø	ILLEGAL1	W,BXC		SAVE DATA IN B-REG
247	Ø364		Ø 1ØØ Ø1Ø ØØØ		ROM 2		GO TO ROM 2 TO PRINT MESSAGE
248	Ø365		Ø ØØØ ØØØ ØØØ		DUMMY		
249	Ø366		Ø Ø1Ø 1Ø1 ØØØ	START1	CXM		PREPARE TO MODIFY SELECT CODE
25Ø	Ø367		1 ØØ1 ØØ1 1ØØ		PT9		SET POINTER TO SELECT CODE DIGIT
251	Ø37Ø		Ø 111 1ØØ Ø1Ø		P,CP1C		INCREMENT SELECT CODE
252	Ø371	Ø373	1 111 1Ø1 111		BRN	PRINTSC	IF NO CARRY PRINT SELECT CODE
253	Ø372		Ø 111 1ØØ Ø1Ø		P,CP1C		CARRY ! INCREMENT ONCE MORE
254	Ø373		Ø Ø1Ø 1Ø1 ØØØ	PRINTSC	CXM		RESTORE DATA
255	Ø374	Ø363	1 111 ØØ1 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A. (FCTN BLK KEY)
256	Ø375	Ø363	1 111 ØØ1 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
257	Ø376	Ø363	1 111 ØØ1 111		BRN	ILLEGAL1	ILLEGAL KEY CODE OR S.A.
258	Ø377	ØØ15	Ø ØØØ 11Ø 111	NO MEMORY	BRN	XMEM	TRIED TO STEP OR RUN W/NO MEMORY, GO TO ERROR

TYPEWRITER INTERFACE LISTING (ROM 1)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		1 010 100 000	LISTM	SBL		SET BUSY LIGHT
4	0001		0 000 010 100	LISTK	YS0		IS KEY STILL DOWN?
5	0002	0357	1 110 111 111		BRN	LIST1	NO, CONTINUE ROUTINE
6	0003		0 000 100 100		RS0		YES, RESET STATUS BIT AND TEST KEY DOWN AGAIN
7	0004	0001	0 000 000 111		BRN	LISTK	TEST FOR KEY DOWN
8	0005		0 101 110 010	SM1	WP,CM1C		SUBTRACT 1 FROM SPACES, WAS SPACES=0?
9	0006	0011	0 000 100 111		BRN	SM1F	NO, RETURN TO CALLING PROGRAM
10	0007		0 110 010 000	FTS	ROM 3		YES, GO TO ROM 3 FOR FIELD TOO SMALL ROUTINE
11	0010	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
12	0011		0 000 110 000	SM1F	RETURN		RETURN TO CALLING ROUTINE
13	0012	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
14	0013	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
15	0014	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
16	0015		0 000 000 000		DUMMY		
17	0016	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
18	0017		0 110 010 000	POCTFLG2	ROM 3		GO TO ROM 3 TO TEST FOR CODE "377"
19	0020	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
20	0021		0 100 010 000	SYNTAX	ROM 2		GO TO ROM 2 TO PRECESS SYNTAX
21	0022	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
22	0023	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
23	0024		0 000 010 000	STEP1	ROM 0		GO TO ROM 0 TO PROCESS STEP
24	0025		0 011 010 000		TKRA		ACCEPT FUNCTION BLOCK KEYCODE
25	0026	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
26	0027		0 000 000 000		DUMMY		
27	0030	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
28	0031		0 000 000 000		DUMMY		
29	0032	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
30	0033	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
31	0034	0373	1 111 101 111		BRN	XKEY1	INVALID KEYCODE
32	0035		0 000 000 000		DUMMY		
33	0036		1 000 101 110	LIST	W,BXC		SAVE DATA IN B-REG
34	0037	0000	0 000 000 011		BRN	LISTM	CONTINUE ROUTINE
35	0040		1 000 010 000	RECALLD	ROM 4		GO TO ROM 4 TO CONTINUE ROUTINE
36	0041		0 101 101 010	MASK3	X,CM1C		BLANK EXPONENT PART OF WORD
37	0042		1 000 101 110		W,BXC		DATA BACK TO C-REG, MASK TO B-REG
38	0043		0 110 010 100		YS6		IS DATA IN SCIENTIFIC NOTATION?
39	0044	0073	0 011 101 111		BRN	EXIT1	NO, PREPARE TO EXIT
40	0045		0 000 101 010		X,ZTB		YES, UNBLANK EXPONENT MASK
41	0046		0 110 111 010		XS,ZMC		IS EXPONENT POSITIVE?
42	0047	0075	0 011 110 111		BRN	EXIT	YES, PREPARE TO EXIT
43	0050		0 001 001 100	COMPL	PT1		NO, SET POINTER FOR EXPONENT COMPLEMENT
44	0051		0 010 110 010		WP,ZMCC		COMPLEMENT EXPONENT
45	0052		1 110 110 010		WP,AXC		ORIGINAL EXPONENT TO C-REG, COMPL. TO A-REG
46	0053	0075	0 011 110 111		BRN	EXIT	PREPARE TO EXIT
47	0054		0 011 011 110	DSHIFT2	S,ZTC		SET SIGN TO ZERO BEFORE SHIFT
48	0055		1 001 001 110		W,SRC		SHIFT ONCE
49	0056	0222	1 001 001 101		JSB	DSHIFT1	CHECK IF SHIFT IS COMPLETE
50	0057		1 000 101 110	STORE1	W,BXC		SAVE C-REG IN B-REG
51	0060		1 010 101 000		MTC		RECALL TFLAG FOR OP LEVEL TEST
52	0061		0 101 111 110		S,CM1C		DOES OP LEVEL = 0?

TYPEWRITER INTERFACE LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0324	1 101 010 011		BRN	STORE2	NO, CONTINUE TO TEST OP LEVEL
54	0063		1 101 001 100		PT13		YES, PREPARE TO UPDATE OP LEVEL
55	0064		0 101 011 000		LDC5		SET OP LEVEL TO FIVE
56	0065		0 000 000 000		NOP		SOFTWARE DELAY
57	0066		0 010 101 000	DSRET	CXM		STORE NEW FLAG IN M-REG
58	0067		0 011 001 110		W,ZTC		CLEAR C-REG
59	0070		1 010 010 000		ROM 5		GO TO SUPERVISOR
60	0071		0 000 000 000		DUMMY		
61	0072		0 000 000 000		DUMMY		
62	0073		0 110 001 010	EXIT1	X,CTA		STORE EXPONENT IN A-REG FOR PRINT ROUTINE
63	0074		0 110 011 110		S,CTA		LOAD SIGN OF DATA IN DISPLAY
64	0075		1 000 010 100	EXIT	YS8		DID PRINT ROUTINE CALL?
65	0076	0126	0 101 011 011		BRN	MANEX	NO, GO TO SUPERVISOR (IN ROM 0)
66	0077	0132	0 101 101 011		BRN	PRINT12	YES, CONTINUE ROUTINE
67	0100		1 000 000 100	PRINT1	SS8		SET PRINT ROUTINE FLAG
68	0101		1 010 101 000		MTC		RECALL TFLAG FOR OP LEVEL TEST
69	0102		0 101 111 110		S,CM1C		DOES OP LEVEL = 0 ?
70	0103	0040	0 010 000 011		BRN	RECALLD	NO, CONTINUE TEST
71	0104	0165	0 111 010 111		BRN	ADJDATA	YES, GO TO ADJUST DATA ROUTINE
72	0105		0 101 000 000	SHIFT	IS2		SHIFT ROUTINE
73	0106		0 100 001 000		SLT		SHIFT
74	0107		0 100 001 000		SLT		SHIFT
75	0110		0 100 001 000		SLT		SHIFT
76	0111		0 100 001 000		SLT		SHIFT
77	0112		1 001 000 000		IS1		PREPARE TO TEST IF SHIFT IS COMPLETE
78	0113		0 000 011 100		PRS		SHIFT POINTER
79	0114		0 000 101 100		YP0		IS SHIFT COMPLETE
80	0115	0105	0 100 010 111		BRN	SHIFT	NO, CONTINUE SHIFT
81	0116		0 110 010 000		ROM 3		YES, CONTINUE STORE ROUTINE
82	0117		1 000 101 110	DISPLAY1	W,BXC		SAVE DATA IN B-REG
83	0120		1 010 101 000		MTC		PREPARE FOR OP LEVEL TEST
84	0121	0344	1 110 010 011		BRN	DATA?	TEST FOR DATA TO BE DISPLAYED
85	0122		1 001 111 010	PRINT6	XS,AM1		IS EXPONENT POSITIVE?
86	0123	0164	0 111 010 011		BRN	SPOUT	NO, PREPARE TO OUTPUT SPACES
87	0124	0362	1 111 001 101		JSB	SMEXP	YES, SUBTRACT EXPONENT FROM SPACES
88	0125	0164	0 111 010 011		BRN	SPOUT	GO TO SPOUT ROUTINE
89	0126		0 000 101 000	MANEX	DSTO		TURN ON DISPLAY
90	0127		0 101 100 100		RS5		RESET STATUS BIT 5
91	0130		0 110 100 100		RS6		RESET STATUS BIT 6
92	0131		0 000 010 000	DISFIN2	ROM 0		GO TO ROM 0 AND PROCEED IN SUPERVISOR
93	0132		0 100 101 000	PRINT12	CTS		STORE DATA IN STACK
94	0133		1 010 101 000		MTC		RECALL TFLAG FOR FIELD WIDTH
95	0134		0 001 001 100		PT1		SET POINTER FOR SPACE ADJUSTMENT
96	0135		0 001 110 010		WP,CM1		IS WIDTH = 0 ?
97	0136	0141	0 110 000 111		BRN	FNZ	NO, FIELD IS NOT ZERO, CONTINUE ROUTINE
98	0137		0 010 011 000		LDC2		YES, SET FIELD WIDTH TO 20
99	0140		0 001 001 100		PT1		RESET POINTER
100	0141	0004	0 000 010 101	FNZ	JSB	SM1	DECREMENT SPACES BY ONE FOR MSD
101	0142		0 001 111 010		XS,CM1		IS D = 0 ?
102	0143	0367	1 111 011 111		BRN	PRINT9	NO, ADJUST SPACES FOR D NOT EQUAL TO 0
103	0144		1 001 111 110	PRINT2	S,AM1		YES, CONTINUE ROUTINE, IS MANTISSA POSITIVE?
104	0145	0303	1 100 001 111		BRN	PRINT8	NO, ADJUST SPACES FOR MINUS SIGN
105	0146		0 101 010 100	PRINT3	YS5		YES, CONTINUE ROUTINE, DID DATA FIT?
106	0147	0332	1 101 101 011		BRN	PRINT7	YES, CONTINUE ROUTINE
107	0150		0 000 100 110		M,ZTB		NO, CLEAR DECIMAL POINT MASK
108	0151	0362	1 111 001 101		JSB	SMEXP	SUBTRACT EXPONENT FROM SPACES
109	0152		1 001 100 010		P,AM1		IS MOST SIGNIFICANT DIGIT OF EXPONENT = 0 ?
110	0153	0164	0 111 010 011		BRN	SPOUT	NO, GO TO OUTPUR SPACE ROUTINE
111	0154		1 100 001 100		PT12		YES, PREPARE TO LOCATE DECIMAL POINT
112	0155		1 101 101 010	DPADJ1	X,AM1A		DECREMENT EXPONENT, IS ADJUSTMENT COMPLETE?

TYPEWRITER INTERFACE LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156	0263	1 001 001 111		BRN	DPADJ2	NO, ADJUST DECIMAL POINT LOCATION
114	0157		0 111 100 010		P,CP1C		YES, LOAD DECIMAL POINT CODE (1 WILL WORK
115	0150		1 000 100 010		P,BXC		LOAD DECIMAL POINT CODE IN MASK REGISTER
116	0161		1 011 101 010		X,ZTA		CLEAR EXPONENT IN DATA
117	0162		0 000 101 010		X,ZTB		CLEAR MASK IN B-REG
118	0163		0 001 001 100		PT1		INITIALIZE POINTER FOR SPOUT ROUTINE
119	0164		0 110 010 000	SPOUT	ROM 3		GO TO ROM 3 TO FINISH PRINT ROUTINE
120	0165		0 011 001 100	ADJDATA	PT3		SET POINTER FOR SCIENTIFIC NOTATION TEST
121	0166		0 001 100 010		P,CM1		IS FIXED POINT SPECIFIED?
122	0167	0246	1 010 011 011		BRN	SCINOT	NO, GO TO SCIENTIFIC NOTATION ROUTINE
123	0170		1 100 101 110		W,AXB		YES, LOAD DATA IN A-REG
124	0171		0 100 101 110		W,ATB		LOAD DATA IN B-REG
125	0172		0 010 010 110	ADJDATA1	MS,BTC		LOAD MANTISSA IN C-REG FOR ADJUSTMENT
126	0173		0 101 111 010	EXPADJ1	XS,CM1C		DECREMENT RND, CHECK IF CARRY?
127	0174	0242	1 010 001 011		BRN	EXPADJ2	NO CARRY, CONTINUE EXP. ADJUSTMENT
128	0175		0 011 001 010	FP1	X,ZTC		YES, CLEAR C-REG EXP, ELIMINATE FALSE ROUNDUP
129	0176		0 001 001 100		PT1		PREPARE TO TEST MSD OF (EXP + RND)
130	0177		1 001 100 010		P,AM1		IS MOST SIGNIFICANT DIGIT OF EXPONENT = 0 ?
131	0200	0204	1 000 010 011		BRN	EXP1	NO, DATA MAY NOT FIT
132	0201		1 001 111 010		XS,AM1		YES, CONTINUE TEST, IS EXP POSITIVE?
133	0202	0217	1 000 111 111		BRN	EXP3	NO, CONTINUE TEST
134	0203	0223	1 001 001 111		BRN	DSHIFT1	YES, DATA WILL FIT, GO TO SHIFT ROUTINE
135	0204		1 001 111 010	EXP1	XS,AM1		IS EXPONENT POSITIVE?
136	0205	0210	1 000 100 011		BRN	EXP2	NO, CONTINUE TEST
137	0206		0 101 000 100	DTL	SS5		YES DATA WILL NOT FIT, SET STATUS BIT
138	0207	0246	1 010 011 011		BRN	SCINOT	GO TO SCIENTIFIC NOTATION ROUTINE
139	0210		1 111 101 010	EXP2	X,AP1A		IS EXP = 999?
140	0211	0215	1 000 110 111		BRN	DTS	NO, DATA IS TOO SMALL
141	0212		0 011 011 110		S,ZTC		YES, CLEAR SIGN FOR SHIFT
142	0213		1 001 001 110		W,SRC		SHIFT DATA ONCE (WILL SHIFT 9 MORE TIMES
143	0214	0222	1 001 001 101		JSB	DSHIFT1	GO TO SHIFT ROUTINE
144	0215		0 011 001 110	DTS	W,ZTC		CLEAR C-REG (DISPLAY = 0)
145	0216	0222	1 001 001 101		JSB	DSHIFT1	GO TO DATA SHIFT ROUTINE
146	0217		1 111 111 010	EXP3	XS,AP1A		IS EXPONENT NEGATIVE?
147	0200	0206	1 000 011 011		BRN	DTL	NO, DATA TOO LARGE
148	0221		1 101 111 010		XS,AM1A		YES, RESTORE SIGN OF EXPONENT
149	0222	0214	1 000 110 101		JSB	DTS	BUT DATA IS TOO SMALL TO BE DISPLAYED
150	0223		0 000 001 100	DSHIFT1	PT0		PREPARE POINTER FOR DATA SHIFT
151	0224		1 111 100 010		P,AP1A		DOES LSD OF EXPONENT=9? (IS SHIFT COMPLETE?)
152	0225	0054	0 010 110 011		BRN	DSHIFT2	NO, SHIFT DATA ONCE
153	0226		1 011 101 110	DSHIFT3	W,ZTA		PREPARE A-REG FOR ROUNDUP
154	0227		0 110 001 010		X,CTA		LOAD GUARD DIGIT IN A-REG
155	0230		1 111 001 110		W,APCA		ROUNDUP
156	0231		1 010 101 000		MTC		GET ROUND DIGIT FOR DECIMAL POINT LOCATION
157	0232		0 010 001 100		PT2		INITIALIZE DECIMAL POINT LOCATION
158	0233		0 000 111 100	DSHIFT4	PLS		ADJUST DECIMAL POINT LOCATION
159	0234		0 101 111 010		XS,CM1C		IS RND = 0 ?
160	0235	0233	1 001 101 111		BRN	DSHIFT4	NO, CONTINUE DECIMAL POINT ADJUSTMENT
161	0236		0 011 101 100		YP3		YES, WAS ORIGINAL RND = 0 ?
162	0237	0252	1 010 101 011		BRN	MASK	NO, LOAD DECIMAL POINT
163	0240		0 011 001 110		W,ZTC		YES, CLEAR C-REG FOR MASK GENERATION
164	0241	0253	1 010 110 001		JSB	MASK1	DO NOT LOAD DECIMAL POINT
165	0242		1 111 101 010	EXPADJ2	X,AP1A		INCREMENT EXPONENT
166	0243	0172	0 111 101 101		JSB	EXPADJ1	CHECK IF EXP. ADJUSTMENT IS COMPLETE
167	0244		1 101 010 010	SMEXP1	WP,AMCA		RESTORE EXPONENT IN A-REG
168	0245		0 000 110 000		RETURN		GO TO CALLING ROUTINE

TYPEWRITER INTERFACE LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
169	0246		1 100 001 100	SCINOT	PT12		LOCATE DECIMAL POINT FOR SCIENTIFIC NOTATION
170	0247		0 110 000 100		SS6		SET SCIENTIFIC NOTATION FLAG
171	0250		1 100 101 110		W,AXB		DATA TO A-REG
172	0251		0 100 101 110		W,ATB		DATA TO B-REG
173	0252		0 011 001 110	MASK	W,ZTC		CLEAR C-REG FOR MASK GENERATION
174	0253		0 010 011 000		LDC2		LOAD DECIMAL POINT CODE
175	0254		1 100 001 100	MASK1	PT12		BEGIN SEARCH FOR LEADING ZEROES
176	0255		1 001 100 010	MASK2	P,AM1		IS THIS DIGIT 0?
177	0256	0041	0 010 000 111		BRN	MASK3	NO, TEST OVER
178	0257		0 001 100 010		P,CM1		YES, IS DECIMAL POINT AT THIS LOCATION?
179	0260	0041	0 010 000 111		BRN	MASK3	YES, END OF SEARCH FOR LEADING ZEROES
180	0261		1 001 011 000		LDC9		NO, BLANK THIS DIGIT
181	0262	0265	1 011 010 111		BRN	MASK4	CONTINUE ROUTINE
182	0263		0 000 011 100	DPADJ2	PRS		ADJUST POINTER AND CHECK EXPONENT AGAIN
183	0264	0155	0 110 110 111		BRN	DPADJ1	CHECK EXPONENT
184	0265		0 011 101 100	MASK4	YP3		IS THIS THE LAST DIGIT TO BE BLANKED?
185	0266	0255	1 010 110 111		BRN	MASK2	NO, CONTINUE SEARCH FOR LEADING ZEROES
186	0267	0041	0 010 000 111		BRN	MASK3	YES, CONTINUE ROUTINE
187	0270		0 000 000 000		DUMMY		
188	0271		0 000 000 000		DUMMY		
189	0272		0 000 000 000		DUMMY		
190	0273		0 000 000 000		DUMMY		
191	0274		0 000 000 000		DUMMY		
192	0275		0 000 000 000		DUMMY		
193	0276		0 000 000 000		DUMMY		
194	0277		0 010 101 000	POCTFLG	CXM		RESTORE DATA AND FLAG
195	0300		1 000 101 110		W,BXC		SAVE DATA IN B-REG
196	0301		0 111 110 000		CTT		CLEAR T-REG
197	0302	0306	1 100 011 011		BRN	POCTFLG1	CONTINUE ROUTINE
198	0303	0004	0 000 010 101	PRINT8	JSB	SM1	SUBTRACT 1 FROM SPACES (FOR MINUS SIGN)
199	0304	0146	0 110 011 011		BRN	PRINT3	CONTINUE ROUTINE
200	0305		0 000 000 000		DUMMY		
201	0306		0 101 000 000	POCTFLG1	IS2		PREPARE TO LOAD DATA IN T-REG
202	0307		0 100 101 000		IXT		LOAD DATA IN T-REG
203	0310		1 001 000 000		IS1		PREPARE TO CHECK FOR "377"
204	0311		1 101 001 100		PT13		SET POINTER FOR "377" MASK
205	0312		1 111 011 000		LDC15		LOAD CODE
206	0313		0 000 000 000		NOP		FOR "377"
207	0314		1 111 011 000		LDC15		MASK
208	0315	0017	0 000 111 111		BRN	POCTFLG2	CONTINUE ROUTINE
209	0316		1 010 101 000	RECALL2	MTC		RECALL TFLAG FOR OP LEVEL TEST
210	0317		0 101 111 110		S,CM1C		DOES OP LEVEL = 0?
211	0320	0040	0 010 000 011		BRN	RECALLD	NO, CONTINUE TEST
212	0321		1 101 001 100		PT13		YES, SET POINTER TO UPDATE OP LEVEL
213	0322		0 100 011 000		LDC4		SET OP LEVEL TO FOUR
214	0323	0066	0 011 011 011		BRN	DSRET	PREPARE TO RETURN TO SUPERVISOR
215	0324		1 010 100 100	STORE2	RS10		RESET STATUS BIT
216	0325		0 101 111 110		S,CM1C		DOES OP LEVEL = 1?
217	0326		0 101 111 110		S,CM1C		DOES OP LEVEL = 2?
218	0327	0347	1 110 011 111		BRN	STORE3	NO, CONTINUE TEST
219	0330		1 100 001 100		PT12		YES, INITIALIZE POINTER FOR SHIFT
220	0331	0105	0 100 010 111		BRN	SHIFT	CONTINUE ROUTINE
221	0332		0 110 010 100	PRINT7	YS6		WAS SCIENTIFIC NOTATION SPECIFIED?
222	0333	0122	0 101 001 011		BRN	PRINT6	NO, CONTINUE TEST
223	0334		0 101 100 010		P,CM1C		YES, SUBTRACT TEN FROM SPACES, ARE SPACES < 10?
224	0335	0337	1 101 111 111		BRN	PRINT71	NO, CONTINUE TO SUBTRACT 12 FROM SPACES
225	0336	0007	0 000 011 111		BRN	FTS	YES, GO TO FIELD TOO SMALL ROUTINE
226	0337	0004	0 000 010 101	PRINT71	JSB	SM1	SUBTRACT 1 FROM SPACES (11)
227	0340	0004	0 000 010 101		JSB	SM1	SUBTRACT 1 FROM SPACES (12)

TYPEWRITER INTERFACE LISTING (ROM 1) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
228	0341	0004	0 000 010 101		JSB	SM1	SUBTRACT 1 FROM SPACES (13)
229	0342	0004	0 000 010 101		JSB	SM1	SUBTRACT 1 FROM SPACES (14)
230	0343	0164	0 111 010 011		BRN	SPOUT	GO TO SPOUT ROUTINE
231	0344		0 101 010 100	DATA?	YS5		IS DATA ENTRY FLAG SET
232	0345	0165	0 111 010 111		BRN	ADJDATA	NO, GO TO DATA ADJUSTMENT ROUTINE
233	0346	0350	1 110 100 011		BRN	YENTRY	YES, DISPLAY DATA ENTRY
234	0347		1 000 010 000	STORE3	ROM 4		CONTINUE TEST
235	0350		1 100 101 000	YENTRY	DNR		RECALL DATA FROM STACK
236	0351		0 100 101 000		CTS		RESTORE STACK
237	0352		1 110 101 110		W,AXC		LOAD DATA IN A-REG
238	0353		0 100 001 110	LREG	W,SLA		SHIFT DATA TO PROPER POSITION
239	0354		0 100 001 110		W,SLA		SHIFT DATA TO PROPER POSITION
240	0355		0 100 001 110		W,SLA		SHIFT DATA TO PROPER POSITION
241	0356	0254	1 010 110 011		BRN	MASK1	FINISH DISPLAY WITH MASK
242	0357		1 010 101 000	LIST1	MTC		RECALL TFLAG FOR OP LEVEL TEST
243	0360		0 101 111 110		S,CM1C		DOES OP LEVEL=0?
244	0361	0021	0 001 000 111		BRN	SYNTAX	NO, GO TO SYNTAX ERROR ROUTINE
245	0362		1 010 010 000	LIST2	ROM 5		GO TO ROM 5 TO CONTINUE LIST
246	0363		1 110 110 010	SMEXP	WP,AXC		EXCHANGE EXPONENT AND SPACES
247	0364		0 101 010 010		WP,AMCC		SUBTRACT EXP FROM SPACES, IS SPACE <0?
248	0365	0244	1 010 010 011		BRN	SMEXP1	NO, PREPARE TO RETURN
249	0366	0007	0 000 011 111		BRN	FTS	YES, GO TO FIELD TOO SMALL ROUTINE
250	0367	0004	0 000 010 101	PRINT9	JSB	SM1	SUBTRACT 1 FROM SPACES (FOR D NOT EQUAL TO 0)
251	0370		0 101 111 010		XS,CM1C		DECREMENT RND DIGIT, IS ADJUSTMENT COMPLETE?
252	0371	0367	1 111 011 111		BRN	PRINT9	NO, CONTINUE TO ADJUST SPACES
253	0372	0144	0 110 010 011		BRN	PRINT2	YES, CONTINUE ROUTINE
254	0373		0 000 010 000	XKEY1	ROM 0		GO TO ERROR MESSAGE ROUTINE
255	0374		0 010 101 000	DISLREG	CXM		STORE NEW OP LEVEL
256	0375		0 011 001 110		W,ZTC		CLEAR C-REG
257	0376	0352	1 110 101 101		JSB	LREG	DISPLAY LAST REGISTER USED
258	0377		0 000 000 000		DUMMY		

(190)

TYPEWRITER INTERFACE LISTING (ROM 2)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		1 101 110 000			TCS		COMPARE T-REG
4	0001		0 000 111 100			PLS		SHIFT POINTER
5	0002		1 100 101 100			YP12		INTERROGATE POINTER
6	0003	0376	1 111 111 011			BRN	FLG1	TEST FLAG
7	0004	0212	1 000 101 101			JSB	FLAG	TEST IF PRINTER FLAG HAS RETURNED
8	0005		0 011 110 000			ADV		ADVANCE PAPER
9	0006		1 110 101 110			W,AXC		PREPARE TO RESTORE T-REG
10	0007		0 111 110 000			CTT		RESTORE T-REG
11	0010		0 010 101 000			CXM		PREPARE TO SET ZERO FOR OP MODE
12	0011		0 000 011 000			LDC0		SET OP MODE TO 0
13	0012	0057	0 010 111 111			BRN	LDC0	CLEAR SHIFT DIGIT
14	0013		1 000 101 110		LERR	W,BXC		SAVE DATA IN B-REG
15	0014	0232	1 001 101 011			BRN	LERROR3	GO TO ERROR MESSAGE ROUTINE
16	0015	0235	1 001 110 111			BRN	VSPACEF	CONTINUE ROUTINE
17	0016	0013	0 000 101 111			BRN	LERR	NO MEMORY
18	0017		1 000 101 110		CPAR1	W,BXC		SAVE C-REG IN B-REG
19	0020		1 010 101 000			MTC		RECALL TFLAG FOR OP LEVEL TEST
20	0021	0352	1 110 101 011			BRN	CPAR5	CONTINUE ROUTINE
21	0022	0206	1 000 011 011			BRN	SYNTAX	GO TO SYNTAX ERROR ROUTINE
22	0023	0037	0 001 111 111		DECPT1	BRN	DECPT2	GO TO DECPT ROUTINE
23	0024		1 000 101 110		DIGIT1	W,BXC		SAVE DATA IN B-REG
24	0025		1 010 101 000			MTC		RECALL TFLAG FOR OP LEVEL TEST
25	0026	0065	0 011 010 111			BRN	DIGIT2	CONTINUE ROUTINE
26	0027		1 000 101 110		OPAR1	W,BXC		SAVE DATA IN B-REG
27	0030		1 010 101 000			MTC		RECALL TFLAG FOR OP LEVEL TEST
28	0031		0 101 111 110			S,CM1C		DOES OP LEVEL = 0?
29	0032	0211	1 000 100 111			BRN	OPAR2	NO, CONTINUE TEST
30	0033		0 010 111 110			S,ZMCC		YES, SET OP LEVEL TO 1
31	0034		0 111 111 110			S,CP1C		SET OP LEVEL TO 2
32	0035		0 010 101 000			CXM		STORE NEW OP LEVEL
33	0036	0212	1 000 101 011			BRN	CANCEL	GO TO CANCEL ENTRY ROUTINE
34	0037		1 000 101 110		DECPT2	W,BXC		SAVE DATA IN B-REG
35	0040		1 010 101 000			MTC		RECALL TFLAG FOR OP LEVEL TEST
36	0041		0 101 111 110			S,CM1C		DOES OP LEVEL = 0?
37	0042		0 101 111 110			S,CM1C		DOES OP LEVEL = 1?
38	0043	0051	0 010 100 111			BRN	DECPT3	NO, CONTINUE TEST
39	0044		0 110 101 000			STA		YES, BRING STACK DOWN TO A-REG FOR WIDTH UPDATE
40	0045		1 010 101 000			MTC		RECALL TFLAG FOR WIDTH UPDATE
41	0046		0 001 001 100			PT1		SET POINTER FOR WIDTH UPDATE
42	0047		1 110 110 010			WP,AXC		LOAD NEW WIDTH IN TFLAG
43	0050	0331	1 101 100 111			BRN	DECPT4	CONTINUE ROUTINE
44	0051		0 101 111 110		DECPT3	S,CM1C		DOES OP LEVEL = 2?
45	0052		0 101 111 110			S,CM1C		DOES OP LEVEL = 3?
46	0053	0206	1 000 011 011			BRN	SYNTAX	NO, GO TO SYNTAX ERROR
47	0054		0 011 011 110			S,ZTC		YES, SET OP LEVEL TO 0
48	0055		0 011 001 100			PT3		SET POINTER FOR SCIENTIFIC NOTATION
49	0056		0 001 011 000			LDC1		SET SCIENTIFIC NOTATION
50	0057		0 000 011 000		LDC0	LDC0		SET DISPLAY TO 0 (OR CLEAR SHIFT DIGIT)
51	0060		0 010 101 000		DIGEX1	CXM		STORE NEW TFLAG
52	0061		1 000 101 110		DIGEX2	W,BXC		RESTORE DATA IN C-REG

TYPEWRITER INTERFACE LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
53	0062		1 011 101 110		W,ZTA		CLEAR A-REG	
54	0063		0 000 101 110		W,ZTB		CLEAR B-REG	
55	0064		0 000 010 000	CPAR4	ROM 0		GO TO SUPERVISOR IN ROM 0	
56	0065		0 101 111 110	DIGIT2	S,CM1C		DOES OP LEVEL = 0?	
57	0066	0075	0 011 110 111		BRN	DIGENT2	NO, CONTINUE TEST	
58	0067		0 010 111 110		S,ZMCC		YES, SET OP LEVEL TO 1	
59	0070		0 010 101 000		CXM		STORE NEW TFLAG	
60	0071		1 110 101 110	STACK	W,AXC		PUT A-REG CONTENTS IN C-REG	
61	0072		0 101 000 100		SS5		SET DATA ENTRY FLAG	
62	0073		0 100 101 000		CTS		STORE DATA IN STACK	
63	0074	0061	0 011 000 111		BRN	DIGEX2	RETURN TO SUPERVISOR	
64	0075		0 101 111 110	DIGENT2	S,CM1C		DOES OP LEVEL = 1 ?	
65	0076	0105	0 100 010 111		BRN	DIGENT3	NO, CONTINUE TEST	
66	0077		1 100 101 000	STADIG	DNR		YES, BRING STACK DOWN TO UPDATE NUMBER	
67	0100		1 110 101 110		W,AXC		NEW DIGIT TO C-REG, OLD DATA TO A-REG	
68	0101		0 100 001 010		X,SLA		SHIFT OLD DATA LEFT TO ACCEPT NEW DIGIT	
69	0102		0 000 001 100		PT0		SET POINTER FOR UPDATE	
70	0103		0 110 000 010		P,CTA		STORE NEW DIGIT	
71	0104	0071	0 011 100 111		BRN	STACK	STORE RESULT IN STACK AND RETURN TO SUPERVISOR	
72	0105		0 101 111 110	DIGENT3	S,CM1C		DOES OP LEVEL = 2?	
73	0106	0117	0 100 111 111		BRN	DIGENT4	NO, CONTINUE TEST	
74	0107		1 110 101 110	BCODE	W,AXC		YES, LOAD DIGIT IN C-REG	
75	0110		0 101 000 000		IS2		PREPARE TO BUILD CODE IN T-REG	
76	0111		0 100 001 000		SLT		SHIFT PREVIOUS CODE	
77	0112		0 100 001 000		SLT		TO MAKE ROOM FOR	
78	0113		0 100 001 000		SLT		NEXT OCTAL DIGIT	
79	0114		0 011 001 000		IOR		LOAD NEW DIGIT IN T-REG	
80	0115		1 001 000 000		IS1		PREPARE TO LOAD DIGIT IN DISPLAY DATA	
81	0116	0061	0 011 000 111		BRN	DIGEX2	GO TO EXIT ROUTINE	
82	0117		0 101 111 110	DIGENT4	S,CM1C		DOES OP LEVEL = 3 ?	
83	0120	0130	0 101 100 011		BRN	DIGENT5	NO, CONTINUE ROUTINE	
84	0121		0 011 001 100	RND	PT3		YES, SET POINTER TO LOAD RND DIGIT	
85	0122		0 000 011 000		LDC0		CLEAR SCIENTIFIC NOTATION FLAG	
86	0123		0 100 001 010		X,SLA		POSITION RND DIGIT FOR STORAGE	
87	0124		0 100 001 010		X,SLA		POSITION RTD DIGIT FOR STORAGE	
88	0125		1 110 100 010		P,AXC		LOAD NEW RND DIGIT IN TFLAG	
89	0126		0 011 011 110	SET0	S,ZTC		SET OP LEVEL TO 0	
90	0127	0057	0 011 000 001		JSB	DIGEX1	GO TO EXIT ROUTINE	
91	0130		0 101 111 110	DIGENT5	S,CM1C		DOES OP LEVEL = 4 ?	
92	0131	0313	1 100 101 111		BRN	DIGENT6	NO, CONTINUE TEST	
93	0132		1 001 101 110	SRE	W,AM1		YES, WAS DIGIT ENTERED A ZERO ?	
94	0133	0135	0 101 110 111		BRN	DS1	NO, DO NOT ADJUST DIGIT	
95	0134	0337	1 101 111 111		BRN	DSERR	YES, GO TO ERROR ROUTINE	
96	0135		0 001 001 100	DS1	PT1		INITIALIZE POINTER FOR DIGIT SHIFT	
97	0136		0 100 001 110	SHIFT	W,SLA		SHIFT DIGIT LEFT	
98	0137		0 000 111 100		PLS		SHIFT POINTER LEFT	
99	0140		1 100 101 100		YP12		IS SHIFT COMPLETE ?	
100	0141	0136	0 101 111 011		BRN	SHIFT	NO, CONTINUE TO SHIFT	
101	0142		1 111 101 110		W,AP1A		YES, LOAD EXP. PART OF DATA STORAGE ADDRESS	
102	0143		1 110 101 110		W,AXC		LOAD ADDRESS IN C-REG, FLAG TO A-REG	
103	0144		1 110 011 000		LDC14		FINISH LOADING DATA STORAGE ADDRESS	
104	0145		1 001 110 000		ATDS		ENABLE SELECTED REGISTER	
105	0146		1 000 101 110		W,BXC		BRING DATA BACK TO C-REG	
106	0147		1 111 111 110		S,AP1A		DOES OP LEVEL = 4 ?	
107	0150	0321	1 101 000 111		BRN	DS2	NO, STORE OR EXCHANGE DATA	
108	0151		1 011 111 000		DSTC		YES, RECALL DATA FROM DATA STORAGE	
109	0152		1 011 110 000	DS3	DTDS		LOAD DATA IN DATA STORAGE	
110	0153		1 011 111 110	DS4	S,ZTA		SET OP LEVEL TO 0	
111	0154		1 000 101 110		W,BXC		SAVE DATA IN B-REG	
112	0155		1 110 101 110		W,AXC		RESTORE FLAG IN C-REG	

TYPEWRITER INTERFACE LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156	0060	0 011 000 011		BRN	DIGEX1	RESTORE ALL DATA AND RETURN TO SUPERVISOR
114	0157		1 000 100 100	XNOTE5	RS8		RESET PRINT FLAG
115	0160		0 111 111 000		TTC		PREPARE TO SAVE T-REG CONTENTS
116	0161		0 110 001 110		W,CTA		SAVE IN A-REG
117	0162		1 101 001 100		PT13		GENERATE MESSAGE FOR PRINTOUT
118	0163		1 010 101 000		MTC		RECALL TFLAG FOR CHANNEL NUMBER
119	0164		1 111 011 000	BLANK	LDC15		LOAD CODE FOR BLANK
120	0165		1 001 101 100		YP9		IS POINTER AT DIGIT 9 ?
121	0166	0164	0 111 010 011		BRN	BLANK	NO, CONTINUE LOADING BLANKS
122	0167		0 000 011 100		PRS		SHIFT POINTER TO SKIP SELECT CODE
123	0170		1 111 011 000		LDC15		X WHERE X = CURRENT SELECT CODE
124	0171		1 100 011 000		LDC12		
125	0172		1 111 011 000		LDC15		G M PN
126	0173		0 000 011 000		LDC0		E E R
127	0174		1 111 011 000		LDC15		N S I O
128	0175		1 011 011 000		LDC11		E S O N
129	0176		1 111 011 000		LDC15		R A N T T
130	0177		1 011 011 000		LDC11		A G E
131	0200		1 111 011 000		LDC15		T E R E
132	0201		0 111 110 000		CTT		E
133	0202		1 101 001 100		PT13		5
134	0203		0 101 011 000		LDC5		X WHERE X = 0 – 4
135	0204		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
136	0205	0263	1 011 010 001	SYNTAX1	JSB	RS89	RESET STATUS BITS
137	0206	0156	0 110 111 101	SYNTAX	JSB	XNOTE5	BEGIN NOTE GENERATION
138	0207		0 000 011 000		LDC0		FINISH NOTE
139	0210	0372	1 111 101 011		BRN	PRINTER	GO TO PRINT ROUTINE
140	0211		1 000 010 000	OPAR2	ROM 4		GO TO ROM 4 TO CONTINUE TEST
141	0212		0 000 010 000	CANCEL	ROM 0		GO TO ROM 0 TO PROCESS CANCEL ENTRY
142	0213		1 011 010 100	FLAG	YS11		HAS FLAG RETURNED ?
143	0214	0213	1 000 101 111		BRN	FLAG	NO, CONTINUE TO TEST FLAG
144	0215		1 011 100 100		RS11		YES, RESET FLAG STATUS BIT
145	0216		0 000 110 000		RETURN		RETURN TO PRINTER ROUTINE
146	0217		0 101 000 000	TEST	IS2		TEST DATA ACCORDING TO MASK
147	0220		0 010 001 000		XOR		PREPARE TEST RESULTS
148	0221		1 101 001 000		TDEC		DECREMENT TEST RESULTS
149	0222		1 110 001 000		YBC		WAS THERE A CARRY ?
150	0223		1 110 001 000		YBC		WAS THERE A CARRY ?
151	0224		1 100 001 000		TINC		RESTORE TEST RESULTS
152	0225		0 010 001 000		XOR		RESTORE ORIGINAL DATA
153	0226		1 001 000 000		IS1		PREPARE TO RETURN
154	0227		0 011 001 110		W,ZTC		CLEAR C-REG
155	0230		0 000 000 000		NOP		ELIMINATE CARRY
156	0231	0317	1 100 111 111		BRN	RETURN	RETURN TO ROM 3 TO CONTINUE STORE ROUTINE
157	0232	0156	0 110 111 101	LERROR3	JSB	XNOTE5	BEGIN NOTE GENERATION
158	0233		0 011 011 000		LDC3		FINISH NOTE
159	0234	0372	1 111 101 011		BRN	PRINTER	GO TO PRINTER ROUTINE
160	0235		0 101 000 000	VSPACEF	IS2		PREPARE TO LOAD LF CODE
161	0236	0001	0 000 001 001		LIO	004	LOAD LF CODE
162	0237		1 001 000 000		IS1		PREPARE TO OUTPUT CODE
163	0240		1 000 101 110	MP	W,BXC		SAVE DATA IN B-REG
164	0241		1 010 101 000		MTC		RECALL TFLAG FOR OP LEVEL TEST
165	0242		0 101 111 110		S,CM1C		DOES OP LEVEL = 0?
166	0243		0 101 111 110		S,CM1C		DOES OP LEVEL = 1 ?
167	0244	0205	1 000 010 111		BRN	SYNTAX1	NO, GO TO SYNTAX ERROR ROUTINE
168	0245		0 011 011 110		S,ZTC		YES, SET OP LEVEL TO 0
169	0246		0 010 101 000		CXM		STORE NEW TFLAG
170	0247		0 101 000 000		IS2		PREPARE TO LOAD CODE IN OUTPUT POSITION
171	0250		0 101 101 000		SRI		LOAD CODE IN OUTPUT POSITION
172	0251		1 001 000 000		IS1		PREPARE TO RECALL DATA FROM STACK

TYPEWRITER INTERFACE LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 110 101 000		STA		BRING DATA DOWN FROM STACK
174	0253		0 011 001 110		W,ZTC		CLEAR C-REG
175	0254		0 111 110 000		CTT		CLEAR T-REG
176	0255		0 010 001 110		W,BTC		SAVE DATA IN C AND B-REG
177	0256		0 100 101 000		CTS		SAVE DATA IN STACK ALSO
178	0257		1 101 101 110	MP2	W,AM1A		DECREMENT COUNTER, IS OUTPUT FINISHED ?
179	0260	0272	1 011 101 011		BRN	WAIT	NO, OUTPUT FUNCTION
180	0261	0262	1 011 001 101		JSB	MP5	YES, EXIT FROM ROUTINE
181	0262	0061	0 011 000 111		BRN	DIGEX2	EXIT FROM ROUTINE
182	0263		1 100 101 000	MP5	DNR		RECALL DATA FROM STACK
183	0264		1 000 100 100	RS89	RS8		RESET STATUS BIT 8
184	0265		1 001 100 100		RS9		RESET STATUS BIT 9
185	0266		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
186	0267	0217	1 000 111 111		BRN	TEST	GO TO TEST ROUTINE
187	0270		0 001 011 000	ILLEGAL	LDC1		FINISH NOTE FOR ILLEGAL KEY CODE OR S.A.
188	0271	0372	1 111 101 011		BRN	PRINTER	PRINT ERROR MESSAGE
189	0272		0 110 010 000	WAIT	ROM 3		GO TO WAIT ROUTINE IN ROM 3
190	0273		1 000 101 110	DS5	W,BXC		PREPARE DATA
191	0274	0153	0 110 101 111		BRN	DS4	CONTINUE ROUTINE
192	0275		1 001 010 100		YS9		IS HSPACEF FLAG SET ?
193	0276	0304	1 100 010 011		BRN	COUNT1	NO, WAIT
194	0277	0257	1 010 111 111		BRN	MP2	YES, OUTPUT SPACE
195	0300		0 101 000 000	HSPACEB	IS2		PREPARE TO LOAD CODE FOR BACKSPACE
196	0301	0003	0 000 010 001		LIO	010	LOAD CODE FOR BACKSPACE
197	0302		1 001 000 000		IS1		PREPARE TO OUTPUT CODE
198	0303	0240	1 010 000 011		BRN	MP	OUTPUT CODE
199	0304		0 011 001 010	COUNT1	X,ZTC		CLEAR COUNTER
200	0305		0 010 001 100		PT2		SET POINTER TO INITIALIZE COUNTER
201	0306		0 101 011 000		LDC5		INITIALIZE COUNTER
202	0307		0 111 101 010	COUNT2	X,CP1C		INCREMENT COUNTER, CARRY?
203	0310	0307	1 100 011 111		BRN	COUNT2	NO, CONTINUE TO COUNT
204	0311	0257	1 010 111 111		BRN	MP2	YES, OUTPUT FUNCTION
205	0312		0 000 000 000		DUMMY		
206	0313		0 101 111 110	DIGENT6	S,CM1C		DOES OP LEVEL = 5 ?
207	0314	0357	1 110 111 111		BRN	DIGENT7	NO, CONTINUE TEST
208	0315		0 111 111 110		S,CP1C		YES, RESTORE OP LEVEL FOR LATER TEST
209	0316	0131	0 101 101 001		JSB	SRE	GO TO DATA STORAGE STORE/RECALL/ EXCHANGE ROUTINE
210	0317		0 110 010 000	RETURN	ROM 3		RETURN TO ROM 3
211	0320	0346	1 110 011 011		BRN	TAB	CONTINUE ROUTINE
212	0321		1 010 010 100	DS2	YS10		IS EXCHANGE FLAG SET ?
213	0322	0152	0 110 101 011		BRN	DS3	NO, GO TO STORE ROUTINE
214	0323		1 010 100 100		RS10		YES, RESET FLAG
215	0324		1 000 101 110		W,BXC		SAVE DATA IN B-REG
216	0325		1 011 111 000		DSTC		RECALL DATA FROM DATA STORAGE
217	0326		1 000 101 110		W,BXC		EXCHANGE DATA
218	0327		1 011 110 000		DTDS		STORE OLD DATA IN DATA STORAGE
219	0330	0273	1 011 101 111		BRN	DS5	CONTINUE ROUTINE
220	0331		1 101 001 100	DECPT4	PT13		SET POINTER FOR OP LEVEL UPDATE
221	0332		0 011 011 000		LDC3		SET OP LEVEL TO 3
222	0333	0060	0 011 000 011		BRN	DIGEX1	RETURN TO SUPERVISOR
223	0334		0 101 100 100	RESET	RS5		RESET DID DATA FIT FLAG
224	0335		0 110 100 100		RS6		RESET SCIENTIFIC NOTATION FLAG
225	0336	0367	1 111 011 111		BRN	RESETD	CONTINUE ROUTINE
226	0337	0156	0 110 111 101	DSERR	JSB	XNOTE5	BEGIN NOTE GENERATION
227	0340		0 010 011 000		LDC2		FINISH NOTE
228	0341	0372	1 111 101 011		BRN	PRINTER	PRINTER ERROR MESSAGE
229	0342		0 101 000 000	HSPACEF	IS2		PREPARE TO LOAD CODE FOR SPACE
230	0343	0000	0 000 000 011		LIO	001	LOAD CODE FOR SPACE
231	0344		1 001 000 000		IS1		PREPARE TO OUTPUT CODE
232	0345	0240	1 010 000 011		BRN	MP	OUTPUT CODE

TYPEWRITER INTERFACE LISTING (ROM 2) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
233	0346		0 101 000 000	TAB	IS2		PREPARE TO LOAD CODE FOR TAB
234	0347	0037	0 010 000 001		LIO	100	LOAD CODE FOR TAB
235	0350		1 001 000 000		IS1		PREPARE TO OUTPUT CODE
236	0351	0240	1 010 000 011		BRN	MP	OUTPUT CODE
237	0352		0 101 111 110	CPAR5	S,CM1C		DOES OP LEVEL=0?
238	0353		0 101 111 110	CPAR2	S,CM1C		DOES OP LEVEL = 1?
239	0354		0 101 111 110		S,CM1C		DOES OP LEVEL = 2?
240	0355	0362	1 111 001 011		BRN	CPAR6	NO, CONTINUE TEST
241	0356	0126	0 101 011 011		BRN	SET0	YES, SET OP LEVEL TO 0
242	0357		0 101 111 110	DIGENT7	S,CM1C		DOES OP LEVEL=6?
243	0360	0077	0 011 111 111		BRN	STADIG	NO, BUILD NUMBER IN STACK
244	0361	0107	0 100 011 111		BRN	BCODE	YES, BUILD OCTAL CODE IN T-REG
245	0362		1 000 010 000	CPAR6	ROM 4		GO TO ROM 4 TO CONTINUE TEST
246	0363	0262	1 011 001 101		JSB	MP5	RECALL DATA AND RESET STATUS BITS
247	0364	0334	1 101 110 011		BRN	RESET	CONTINUE ROUTINE
248	0365	0156	0 110 111 101	ILL	JSB	XNOTE5	BEGIN NOTE GENERATION
249	0366	0270	1 011 100 011		BRN	ILLEGAL	FINISH NOTE GENERATION
250	0367		1 000 101 110	RESETD	W,BXC		SAVE DATA IN B-REG
251	0370	0156	0 110 111 101	FAILURE	JSB	XNOTE5	BEGIN NOTE GENERATION
252	0371		0 100 011 000		LDC4		FINISH NOTE
253	0372		1 111 011 000	PRINTER	LDC15		BEGIN PRINTER ROUTINE
254	0373		1 111 011 000		LDC15		FINISH NOTE GENERATION
255	0374		1 111 001 100		PT15		SET POINTER FOR PRINTER OPERATION
256	0375		0 101 110 000		PRE		ENABLE PRINTER
257	0376	0212	1 000 101 101	FLG1	JSB	FLAG	TEST IF PRINTER FLAG HAS RETURNED
258	0377		1 111 110 000		CCS		COMPARE C-REG

TYPEWRITER INTERFACE LISTING (ROM 3)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
3	0000		0 101 000 000		IS2		PREPARE TO TEST FOR EOL
4	0001	0037	0 010 000 001		LIO	100	LOAD SELECT CODE FOR EOL TEST
5	0002		1 100 100 000		TG9		TEST FOR EOL
6	0003		1 001 000 000		IS1		PREPARE TO TEST RESULT
7	0004		1 011 010 100		YS11		WAS IT EOL?
8	0005	0363	1 111 001 111		BRN	TCA1	YES, OUTPUT TAB CLEAR ALL CODE
9	0006		1 011 100 100		RS11		NO, RESET STATUS BIT
10	0007	0376	1 111 111 011		BRN	TCASPA	CONTINUE TO SPACE
11	0010		1 010 101 000	FTS	MTC		FIELD IS TOO SMALL, RECALL FIELD WIDTH
12	0011		0 101 110 010	FTS1	WP,CM1C		DECREMENT SPACES, IS FIELD = 0 YET?
13	0012	0370	1 111 100 011		BRN	FTS2	NO, CONTINUE TO OUTPUT "$"
14	0013	0353	1 110 101 111		BRN	PREXIT	YES, GO TO EXIT ROUTINE
15	0014		1 000 101 110	OCTFLG3	W,BXC		RESTORE DATA
16	0015		0 000 010 000	1CPAR	ROM 0		GO TO ROM 0 TO ENTER CPAR ROUTINE
17	0016		0 000 011 100	PTADJ2	PRS		ADJUST POINTER
18	0017	0334	1 101 110 011		BRN	PTADJ1	CHECK IF ADJUSTMENT IS COMPLETE
19	0020	0265	1 011 011 001		JSB	TEST	IS THIS CODE "377"?
20	0021		1 011 010 100		YS11		WAS CODE "377"?
21	0022	0014	0 000 110 011		BRN	OCTFLG3	YES, GO TO CPAR ROUTINE
22	0023		1 011 100 100		RS11		NO, RESET STATUS BIT FOR NEXT TEST
23	0024	0117	0 100 111 111		BRN	LFTESTP	GO TO LINE FEED TEST
24	0025		0 000 000 000		DUMMY		
25	0026		0 000 010 000	EXEC	ROM 0		EXECUTE NEXT INSTRUCTION
26	0027		0 011 001 110	FLGEXC1	W,ZTC		CLEAR C-REG FOR ADDRESS
27	0030		0 111 101 110		W,CP1C		LOAD EXPONENT ADDRESS
28	0031		1 100 001 100		PT12		SET POINTER FOR MANTISSA ADDRESS
29	0032		1 110 011 000		LDC14		LOAD MANTISSA PART OF ADDRESS
30	0033		1 001 110 000		ATDS		SELECT CHIP 14B, REGISTER 0
31	0034		1 100 001 100		PT12		START FLAG EXCHANGE, SELECT UPDATE DIGIT
32	0035		1 011 111 000		DSTC		CALL D.S. REG TO C-REG
33	0036		0 110 001 110		W,CTA		STORE FLAG IN A-REG FOR UPDATE
34	0037		1 010 101 000		MTC		RECALL FLAG TO C-REG
35	0040		0 110 000 010		P,CTA		LOAD OP MODE IN NEW FLAG
36	0041	0152	0 110 101 011		BRN	FLGEXC2	CONTINUE ROUTINE
37	0042	0027	0 001 011 111		BRN	FLGEXC1	CONTINUE
38	0043	0256	1 010 111 101	SETCNTL	JSB	IXT	LOAD I/O REG
39	0044	0232	1 001 101 101	SETCNTL1	JSB	WAIT	WAIT FOR FLAG AND OUTPUT DATA
40	0045		0 101 000 000		IS2		PREPARE TO OUTPUT NOP
41	0046	7777	0 000 000 001		LIO	000	LOAD NOP CODE
42	0047		0 101 101 000		SRI		SHIFT NOP TO OUTPUT POSITION
43	0050		1 001 000 000		ISI		PREPARE TO WAIT FOR FLAG
44	0051	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG AND OUTPUT NOP
45	0052	0355	1 110 110 111		BRN	RETURN	PREPARE TO RETURN TO SUPERVISOR
46	0053	0254	1 010 110 101	DATAOUT9	JSB	LOADIGIT	LOAD DIGIT IN I/O-REG
47	0054	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG THEN OUTPUT DATA
48	0055		1 101 111 110		Z,AM1A		RESTORE ORIGINAL DIGIT IN MASK
49	0056		1 001 111 110		S,AM1		IS DECIMAL POINT NEXT?
50	0057	0344	1 110 010 011		BRN	DPOUT1	YES, GO TO DPOUT ROUTINE
51	0060		1 100 101 100	DATAOUT2	YP12		NO, WAS THIS THE LAST DIGIT?
52	0061	0176	0 111 111 011		BRN	SHIFT	NO, EXAMINE NEXT DIGIT

TYPEWRITER INTERFACE LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062		0 101 010 100		YS5		YES, DID DATA FIT?
54	0063	0206	1 000 011 011		BRN	DATAOUT3	YES, TEST FOR OUTPUT MODE (FIXED OR FLOATING POINT)
55	0064		1 011 101 110		W,ZTA		NO, CLEAR A-REG
56	0065		0 000 101 110		W,ZTB		CLEAR B-REG
57	0066		1 100 101 000		DNR		RECALL DATA FROM STACK
58	0067		0 100 101 000		CTS		RESTORE DATA IN STACK
59	0070		1 110 101 110		W,AXC		DATA TO C-REG
60	0071		0 000 001 100		PT0		PREPARE TO LOAD CONSTANT
61	0072		1 001 011 000		LDC9		LOAD 9
62	0073		1 101 001 010		X,AMCA		SUBTRACT 9 FROM EXPONENT
63	0074		1 011 110 110		MS,ZTA		REMAINING PART OF DATA WILL BE ALL ZEROES
64	0075		1 101 101 010	DATAOUT4	X,AM1A		DECREMENT EXPONENT, IS EXP. = 0?
65	0076	0275	1 011 110 111		BRN	ZOUT	NO, PREPARE TO OUTPUT ZEROES
66	0077	0346	1 110 011 011		BRN	DPOUT2	YES, GO TO DECIMAL POINT OUTPUT ROUTINE
67	0100		0 000 010 000	FLGEXC3	ROM 0		RETURN TO CALLING ROUTINE IN ROM 0
68	0101		0 101 000 000	SPACE	IS2		GENERATE CODE FOR SPACE
69	0102	0000	0 000 000 011		LIO	001	LOAD CODE FOR "SPACE"
70	0103		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
71	0104		1 001 000 000		IS1		PREPARE TO CHECK IF OUTPUT IS COMPLETE
72	0105		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
73	0106		1 001 010 100	DSROUT	YS9		NORMAL RETURN?
74	0107	0160	0 111 000 011		BRN	EX	YES, GO TO EXIT
75	0110		0 011 001 110		W,ZTC		CLEAR C-REG
76	0111		0 111 110 000		CTT		CLEAR T-REG
77	0112		1 000 010 000		ROM 4		NO, GO TO ROM 4 TO CONTINUE ROUTINE
78	0113		1 011 101 110	(PRT)	W,ZTA		CLEAR A-REG
79	0114		1 100 101 000		DNR		RECALL DATA FROM STACK
80	0115		1 000 000 100		SS8		SET PRINT KEY HIT FLAG
81	0116		0 010 010 000		ROM 1		GO TO ROM 1 DISPLAY ROUTINE
82	0117		1 000 101 110	LFTESTP	W,BXC		PREPARE TO STORE DATA IN STACK
83	0120		0 100 101 000		CTS		STORE DATA IN STACK
84	0121		0 011 001 110		W,ZTC		CLEAR C-REG FOR TESTS
85	0122		1 100 001 100	LFTEST	PT12		PREPARE POINTER FOR LINE FEED MASK
86	0123		0 100 011 000		LDC4		LOAD LINE FEED TEST MASK
87	0124	0265	1 011 011 001		JSB	TEST	TEST CODE FOR LINE FEED
88	0125		1 011 010 100		YS11		WAS THERE A CARRY DURING TEST?
89	0126	0043	0 010 001 111		BRN	SETCNTL	NO, CODE IS LINE FEED
90	0127		1 011 100 100		RS11		YES, CODE IS NOT LINE FEED
91	0130		1 101 001 100	TCATEST	PT13		SET POINTER TO LOAD TCA TEST
92	0131		1 000 011 000		LDC8		LOAD TAB
93	0132		0 000 000 000		NOP		DELAY FOR LDC
94	0133		0 010 011 000	TCATEST1	LDC2		MASK
95	0134	0265	1 011 011 001		JSB	TEST	TEST CODE TO SEE IF IT IS TAB CLEAR ALL
96	0135		1 011 010 100		YS11		WAS THERE A CARRY DURING TEST?
97	0136	0376	1 111 111 011		BRN	TCASPA	NO, CODE IS TAB CLEAR ALL, OUTPUT SPACE, CHECK EOL
98	0137		1 011 100 100		RS11		YES, CODE IS NOT TAB CLEAR ALL
99	0140		0 111 111 110	PRTEST	S,CP1C		LOAD CODE FOR PRINT TEST
100	0141	0265	1 011 011 001		JSB	TEST	TEST CODE FOR PRINT
101	0142		1 011 010 100		YS11		WAS THERE A CARRY DURING TEST?
102	0143	0113	0 100 101 111		BRN	(PRT)	NO, CODE IS PRINT
103	0144		1 011 100 100		RS11		YES, CODE IS NOT PRINT
104	0145		0 101 000 000	LCIO1	IS2		PREPARE TO LOAD CODE IN I/O-REG
105	0146		0 100 101 000		IXT		LOAD CODE IN I/O-REG
106	0147		1 001 000 000		IS1		PREPARE TO OUTPUT CODE
107	0150	0232	1 001 101 101	LCIO2	JSB	WAIT	WAIT FOR FLAG AND OUTPUT DATA
108	0151	0355	1 110 110 111		BRN	RETURN	PREPARE TO RETURN TO SUPERVISOR
109	0152		1 011 110 000	FLGEXC2	DTDS		STORE OLD FLAG IN D.S.
110	0153		0 011 001 110		W,ZTC		CLEAR C-REG
111	0154		1 110 101 110		W,AXC		NEW FLAG TO C-REG
112	0155		0 010 101 000		CXM		STORE NEW FLAG IN M-REG

TYPEWRITER INTERFACE LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
113	0156	0100	0 100 000 011		BRN	FLGEXC3	RETURN TO CALLING ROUTINE IN ROM 0	
114	0157		0 000 000 000		DUMMY			
115	0160		1 000 100 100	EX	RS8		YES, RESET PRINT KEY FLAG	
116	0161		0 000 010 000	CANCEL	ROM 0		PREPARE TO GO TO CANCEL ENTRY ROUTINE	
117	0162	0302	1 100 001 101	MINUS	JSB	MINUS1	GENERATE CODE FOR MINUS SIGN	
118	0163	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG THEN OUTPUT MINUS SIGN	
119	0164	0171	0 111 100 111		BRN	DATAOUT1	BEGIN OUTPUTING DATA	
120	0165		0 101 110 010	SPOUT	WP,CM1C		DECREMENT SPACES, READY TO OUTPUT DATA?	
121	0166	0263	1 011 001 111		BRN	SPOUT1	NO, CONTINUE TO OUTPUT	
122	0167		1 001 111 110	DATAOUT	S,AM1		YES, BEGIN DATA, OUTPUT ROUTINE, IS NUM. POS.?	
123	0170	0162	0 111 001 011		BRN	MINUS	NO, GO TO ROUTINE TO OUTPUT MINUS SIGN	
124	0171		0 011 001 100	DATAOUT1	PT3		YES, SET POINTER	
125	0172		0 011 011 110	SET9	S,ZTC		CLEAR SIGN DIGIT	
126	0173		0 101 111 110		S,CM1C		SET SIGN DIGIT TO 9	
127	0174		1 100 101 110		W,AXB		STORE DATA IN B-REG, MASK TO A-REG	
128	0175	0202	1 000 001 011		BRN	SHIFT1	SHIFT MASK ONCE TO THE LEFT	
129	0176		0 000 111 100	SHIFT	PLS		SHIFT POINTER ONCE TO THE LEFT	
130	0177		1 100 101 110		W,AXB		PREPARE TO SHIFT DATA	
131	0200		0 100 001 110		W,SLA		SHIFT DATA ONCE TO THE LEFT	
132	0201		1 100 101 110		W,AXB		RESTORE DATA	
133	0202		0 100 001 110	SHIFT1	W,SLA		SHIFT MASK ONCE TO THE LEFT	
134	0203		1 111 111 110		S,AP1A		IS THIS DIGIT BLANKED?	
135	0204	0053	0 010 101 111		BRN	DATAOUT9	NO, OUTPUT THIS DIGIT	
136	0205	0176	0 111 111 011		BRN	SHIFT	YES, SHIFT DATA AND EXAMINE NEXT DIGIT	
137	0206		0 110 010 100	DATAOUT3	YS6		WAS SCIENTIFIC NOTATION SPECIFIED?	
138	0207	0353	1 110 101 111		BRN	PREXIT	NO, GO TO EXIT ROUTINE	
139	0210		1 100 101 000		DNR		YES, RECALL DATA FROM STACK	
140	0211		0 100 101 000		CTS		RESTORE DATA IN STACK	
141	0212		0 101 000 000		IS2		PREPARE TO OUTPUT E	
142	0213	0062	0 011 001 011		LIO	145	LOAD CODE FOR "E"	
143	0214		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION	
144	0215		1 001 000 000		IS1		PREPARE TO OUTPUT CODE	
145	0216	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG AND OUTPUT E	
146	0217		0 001 111 010		XS,CM1		IS EXPONENT POSITIVE?	
147	0220	0271	1 011 100 111		BRN	MINUS2	NO, PREPARE TO OUTPUT MINUS SIGN	
148	0221	0100	0 100 000 101		JSB	SPACE	YES, GENERATE CODE FOR SPACE	
149	0222	0232	1 001 101 101	EXP	JSB	WAIT	WAIT FOR FLAG THEN OUTPUT "SPACE" OR MINUS SIGN	
150	0223		1 100 101 110	EXP1	W,AXB		PREPARE TO SHIFT DATA	
151	0224		0 100 001 110		W,SLA		SHIFT DATA	
152	0225		0 100 001 110		W,SLA		SHIFT DATA	
153	0226		0 110 100 100		RS6		RESET SCIENTIFIC NOTATION FLAG	
154	0227		1 011 001 100		PT11		SET POINTER	
155	0230	0172	0 111 101 011		BRN	SET9	OUTPUT EXPONENT	
156	0231		0 000 000 000		DUMMY			
157	0232		0 000 000 000		DUMMY			
158	0233		0 111 110 000	WAIT	CTT		SAVE CONTENTS OF C-REG IN T-REG	
159	0234		0 011 001 010		X,ZTC		INITIALIZE COUNTER	
160	0235		0 101 000 000	TESTFLG	IS2		PREPARE TO TEST FLAG	
161	0236		0 000 100 000		YINTF		INTERROGATE FLAG STATUS BIT	
162	0237		1 001 000 000		IS1		PREPARE TO CHECK FLAG STATUS	
163	0240		1 011 010 100		YS11		IS PERIPHERAL READY?	
164	0241	0246	1 010 011 011		BRN	OUTPUT	YES, OUTPUT DATA	
165	0242		1 011 100 100		RS11		NO, RESET STATUS BIT AND CHECK FLAG AGAIN	
166	0243		0 111 101 010		X,CP1C		INCREMENT COUNTER, WAS THERE A CARRY?	
167	0244	0235	1 001 110 111		BRN	TESTFLG	NO, TEST FLAG AGAIN	
168	0245	0361	1 111 000 111		BRN	FAILURE	YES, GO TO FAILURE ROUTINE	
169	0246		0 111 111 000	OUTPUT	TTC		RECALL DATA TO C-REG	

TYPEWRITER INTERFACE LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
170	0247		1 010 100 000		SBL		SET BUSY LIGHT
171	0250		0 101 000 000		IS2		PREPARE TO OUTPUT DATA
172	0251	0017	0 001 000 001		LIO	040	LOAD I/O SELECT CODE
173	0252		0 110 100 000		GIOE		OUTPUT DATA
174	0253		1 001 000 000		IS1		PREPARE TO RETURN TO CALLING ROUTINE
175	0254		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
176	0255		0 010 000 110	LOADIGIT	M,BTC		LOAD DIGIT IN C-REG
177	0256		0 111 110 000		CTT		LOAD DIGIT IN T-REG
178	0257		0 101 000 000	IXT	IS2		PREPARE TO LOAD I/O-REG
179	0260		0 100 101 000		IXT		LOAD DIGIT IN I/O-REG
180	0261		1 001 000 000		IS1		PREPARE TO RETURN
181	0262		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
182	0263	0100	0 100 000 101	SPOUT1	JSB	SPACE	GENERATE CODE FOR SPACE
183	0264	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG THEN OUTPUT SPACE
184	0265	0165	0 111 010 111		BRN	SPOUT	TEST IF OUTPUT IS COMPLETE
185	0266		0 100 010 000	TEST	ROM 2		GO TO ROM 2 TO TEST OCTAL DATA
186	0267	0265	1 011 011 001		JSB	TEST	TEST FOR 377
187	0270		1 010 010 000		ROM 5		RETURN TO ROM 5
188	0271	0302	1 100 001 101	MINUS2	JSB	MINUS1	LOAD CODE FOR MINUS SIGN IN I/O-REG
189	0272	0222	1 001 001 011		BRN	EXP	WAIT FOR FLAG
190	0273	0232	1 001 101 101		JSB	WAIT	OUTPUT FUNCTION/NOP
191	0274		0 100 010 000		ROM 2		RETURN TO ROM 2
192	0275		0 101 000 000	ZOUT	IS2		PREPARE TO OUTPUT ZEROES
193	0276	0107	0 100 100 001		LIO	220	LOAD "0" CODE
194	0277		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
195	0300		1 001 000 000		IS1		PREPARE TO OUTPUT 0
196	0301	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG THEN OUTPUT ZERO
197	0302	0075	0 011 110 111		BRN	DATAOUT4	GO TEST IF OUTPUT IS COMPLETE
198	0303		0 101 000 000	MINUS1	IS2		PREPARE TO OUTPUT MINUS SIGN
199	0304	0116	0 100 111 011		LIO	235	LOAD MINUS SIGN CODE
200	0305		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
201	0306		1 001 000 000		IS1		PREPARE TO RETURN
202	0307		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
203	0310		1 101 001 100	PMS6	PT13		SET POINTER TO FINISH MASK
204	0311		1 111 011 000		LDC15		FINISH MASK
205	0312	0256	1 010 111 101		JSB	IXT	LOAD I/O-REG
206	0313	0265	1 011 011 001		JSB	TEST	TEST FOR 377
207	0314		1 011 010 100		YS11		WAS CODE 377?
208	0315	0014	0 000 110 011		BRN	OCTFLG3	YES, GO TO CPAR ROUTINE
209	0316		1 011 100 100		RS11		NO, RESET STATUS BIT
210	0317		1 000 010 000		ROM 4		GO TO ROM 4
211	0320		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
212	0321		1 101 111 110		S,AM1A		DOES OP LEVEL=3?
213	0322		1 101 111 110		S,AM1A		DOES OP LEVEL=4?
214	0323		1 101 111 110		S,AM1A		DOES OP LEVEL=5?
215	0324		1 101 111 110		S,AM1A		DOES OP LEVEL=6?
216	0325	0026	0 001 011 011		BRN	EXEC	NO, EXECUTE NEXT INSTRUCTION
217	0326		0 010 101 000		CXM		YES, RESTORE FLAG AND DATA
218	0327		1 000 101 110		W,BXC		SAVE DATA IN B-REG
219	0330		1 111 011 000		LDC15		LOAD LAST HALF OF MASK
220	0331	0310	1 100 100 011		BRN	PMS6	CONTINUE ROUTINE
221	0332		1 101 001 100	DPOUT3	PT13		SET POINTER FOR FACIT CODE
222	0333		1 001 011 000		LDC9		LOAD FACIT DIGIT CODE (POINTER SET FOR PTADJ)
223	0334		0 101 111 010	PTADJ1	XS,CM1C		DECREMENT RND, IS RND = 0?
224	0335	0016	0 000 111 011		BRN	PTADJ2	NO, GO TO POINTER ADJUSTMENT
225	0336		0 101 000 000	DPOUT4	IS2		YES, PREPARE TO OUTPUT DECIMAL POINT
226	0337	0016	0 000 111 101		LIO	036	LOAD CODE FOR DECIMAL POINT
227	0340		0 101 101 000		SRI		SHIFT DATA TO OUTPUT POSITION
228	0341		1 001 000 000		IS1		PREPARE TO OUTPUT DECIMAL POINT
229	0342	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG THEN OUTPUT
230	0343	0060	0 011 000 011		BRN	DATAOUT2	CONTINUE ROUTINE
231	0344		0 101 010 100	DPOUT1	YS5		DID DATA FIT?
232	0345	0336	1 101 111 011		BRN	DPOUT4	YES, OUTPUT DECIMAL POINT AND RETURN

TYPEWRITER INTERFACE LISTING (ROM 3) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
233	0346		0 101 100 100	DPOUT2	RS5		NO, RESET DID DATA FIT FLAG		
234	0347		0 110 100 100		RS6		RESET SCIENTIFIC NOTATION FLAG		
235	0350		1 010 101 000		MTC		RECALL TFLAG FOR RND DATA		
236	0351		0 001 111 010		XS,CM1		IS ROUND EQUAL TO 0?		
237	0352	0332	1 101 101 011		BRN	DPOUT3	NO, PREPARE TO OUTPUT DECIMAL POINT		
238	0353		0 101 100 100	PREXIT	RS5		YES, RESET DID DATA FIT FLAG		
239	0354		1 110 010 000		ROM 7		GO TO ROM 7 TO CHECK FOR TAB		
240	0355		1 011 101 110	RETURN	W,ZTA		CLEAR A-REG		
241	0356		1 100 101 000		DNR		RESTORE STACK		
242	0357		1 000 101 110		W,BXC		SAVE DATA IN B-REG		
243	0360	0106	0 100 011 011		BRN	DSROUT	CHECK FOR NORMAL RETURN		
244	0361		0 111 111 000	FAILURE	TTC		RECALL DATA TO C-REG		
245	0362		0 100 010 000	FAILURE1	ROM 2		GO TO ROM 2 TO PROCESS FAILURE (EOL)		
246	0363		0 101 000 000	TCA1	IS2		PREPARE TO RELOAD TCA CODE		
247	0364	0100	0 100 000 101		LIO	202	RELOAD TCA CODE		
248	0365		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION		
249	0366		1 001 000 000		ISI		PREPARE TO OUTPUT CODE		
250	0367	0044	0 010 010 011		BRN	SETCNTL1	GO TO OUTPUT ROUTINE		
251	0370		0 101 000 000	FTS2	IS2		CONTINUE TO OUTPUT "$"		
252	0371	0051	0 010 101 001		LIO	124	LOAD CODE FOR "$"		
253	0372		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION		
254	0373		1 001 000 000		ISI		PREPARE TO CHECK IF OUTPUT IS COMPLETED		
255	0374	0232	1 001 101 101		JSB	WAIT	WAIT FOR FLAG THEN OUTPUT "$"		
256	0375	0011	0 000 100 111		BRN	FTS1	TEST IF OUTPUT IS COMPLETED		
257	0376	0100	0 100 000 101	TCASPA	JSB	SPACE	GENERATE CODE FOR SPACE		
258	0377	0232	1 001 101 101		JSB	WAIT	OUTPUT SPACE		

TYPEWRITER INTERFACE LISTING (ROM 4)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
3	0000		0 000 000 000		DUMMY				
4	0001		0 000 000 000		DUMMY				
5	0002		0 000 000 000		DUMMY				
6	0003		0 000 000 000		DUMMY				
7	0004		0 000 000 000		DUMMY				
8	0005		0 000 000 000		DUMMY				
9	0006		0 000 000 000		DUMMY				
10	0007		0 000 000 000		DUMMY				
11	0010		0 000 000 000		DUMMY				
12	0011		1 111 101 010	INCREG2	X,AP1A			INCREMENT REG	
13	0012	0064	0 011 010 011		BRN	AGAIN		DO IT ALL AGAIN	
14	0013	0254	1 010 110 101	DIV	JSB	OPLEV=1?		DOES OP LEVEL1? (RECALL STACK)	
15	0014	0023	0 001 001 111		BRN	SAVE		CONTINUE ROUTINE	
16	0015		1 010 101 000	ALPHA1	MTC			RECALL TFLAG FOR SHIFT DIGIT	
17	0016		1 011 001 100		PT11			SET POINTER TO SHIFT DIGIT	
18	0017		0 110 011 000		LDC6			LOAD SHIFT DIGIT	
19	0020	0374	1 111 110 011		BRN	ALPHA2		CONTINUE ROUTINE	
20	0021	0254	1 010 110 101	EQUAL	JSB	OPLEV=1?		DOES OP LEVEL = 1 ? (RECALL STACK)	
21	0022		1 111 111 110		S,AP1A			SET TYPEWRITER REG INDICATOR	
22	0023	0274	1 011 110 101	SAVE	JSB	ADTEST		TEST ADDRESS	
23	0024	0106	0 100 011 101		JSB	STOAID		SAVE ADDRESS IN D-REG	
24	0025		0 011 001 110		W,ZTC			CLEAR C-REG	
25	0026		0 100 101 000		CTS			SAVE CLEARED REG IN STACK	
26	0027		1 010 101 000		MTC			RECALL TFLAG FOR UPDATE	
27	0030		1 101 001 100		PT13			SET POINTER FOR OP LEVEL OPDATE	
28	0031		0 111 011 000		LDC7			SET OP LEVEL TO 7	
29	0032	0057	0 010 111 111		BRN	DIGEX1		RETURN TO SUPERVISOR	
30	0033		1 010 101 000	NCHAR	MTC			RECALL TFLAG	
31	0034		1 011 001 100		PT11			SET POINTER TO SHIFT DIGIT	
32	0035		0 101 100 010		P,CM1C			WAS THIS THE LAST CHARACTER ?	
33	0036	0374	1 111 110 011		BRN	ALPHA2		NO, CONTINUE IN SAME REG	
34	0037	0266	1 011 011 011		BRN	INCREG1		YES, TEST FOR NEXT REG	
35	0040		0 000 000 000		DUMMY				
36	0041	0341	1 110 001 001	RECALLD	JSB	OPLEV=7?		DOES OP LEVEL=7?	
37	0042		1 100 101 000		DNR			YES, RECALL RELATIVE ADDRESS	
38	0043		0 111 110 000		CTT			SAVE RELATIVE ADDRESS IN T-REG	
39	0044		0 110 101 000		STA			RECALL ADDRESS TO A-REG	
40	0045		1 111 001 010		X,APCA			ADD RELATIVE & ABSOLUTE ADDRESS	
41	0046	0274	1 011 110 101		JSB	ADTEST		CAN REGISTER BE CALLED ?	
42	0047		0 111 111 000		TTC			YES, RECALL RELATIVE ADDRESS	
43	0050		1 101 001 010		X,AMCA			RESTORE STARTING REG ADDRESS	
44	0051		1 110 101 110		W,AXC			EXCHANGE REL/ABS ADDRESS	
45	0052		0 100 001 110		W,SLA			SHIFT RELATIVE	
46	0053		0 100 001 110		W,SLA			ADDRESS TO MANTISSA	
47	0054		0 100 001 110		W,SLA			PORTION OF REG.	
48	0055		1 111 001 110	RECALLE	W,APCA			LOAD SHIFTED RELATIVE ADDRESS	
49	0056	0063	0 011 010 001		JSB	AGAIN		CONTINUE ROUTINE	
50	0057		0 010 101 000	DIGEX1	CXM			STORE NEW T FLAG	
51	0060		0 100 010 000	DIGEX2	ROM 2			GO TO SUPERVISOR	
52	0061		0 001 101 010	TYPE	X,CM1			IS REG 0 CALLED?	

TYPEWRITER INTERFACE LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
53	0062	0166	0 111 011 011		BRN	NOT0	NO, CONTINUE TEST		
54	0063	0333	1 101 101 111		BRN	DSERR	YES, GO TO ERROR ROUTINE		
55	0064	0106	0 100 011 101	AGAIN	JSB	STOAID	STORE COMPACTED ADDRESS IN STACK		
56	0065	0274	1 011 110 101		JSB	ADTEST	ENABLE REG		
57	0066		1 001 000 100		SS9		SET RETURN FLAG		
58	0067		1 010 101 000		MTC		RECALL TFLAG		
59	0070		0 011 011 110		S,ZTC		SET OP LEVEL TO 0		
60	0071		0 010 101 000		CXM		STORE NEW FLAG		
61	0072		1 011 111 000		DSTC		RECALL CONTENTS OF D.S. TO C-REG		
62	0073		1 000 010 100		YSB		ALPHA OR NUMERIC?		
63	0074	0015	0 000 110 111		BRN	ALPHA1	ALPHA, GO TO ALPHA ROUTINE		
64	0075		0 000 010 000	PRT	ROM 0		NUMERIC, GO TO PRINT ROUTINE		
65	0076		1 000 001 100	SLEFT	PT8		SET POINTER FOR SHIFT		
66	0077		0 101 000 000	CONT	IS2		PREPARE TO SHIFT		
67	0100		0 100 001 000		SLT		SHIFT LEFT ONE BIT		
68	0101		1 001 000 000		IS1		PREPARE TO SHIFT POINTER		
69	0102		0 000 011 100		PRS		SHIFT POINTER		
70	0103		0 000 101 100		YP0		IS SHIFT COMPLETE?		
71	0104	0077	0 011 111 111		BRN	CONT	NO, CONTINUE SHIFT		
72	0105		0 000 110 000		RETURN		YES, RETURN TO CALLING ROUTINE		
73	0106		0 110 101 000	RCLDTA	STA		RECALL CONTENTS OF D-REG TO A-REG		
74	0107		1 110 101 110	STOAID	W,AXC		PREPARE TO STORE CONTENTS OF A-REG IN STACK		
75	0110		0 100 101 000		CTS		STORE IN STACK		
76	0111		1 110 101 110		W,AXC		RESTORE REGISTERS		
77	0112		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE		
78	0113	0264	1 011 010 011	OUTD	BRN	TESTPA	RETURN FROM PRINT/ALPHA , OUTPUT MORE ?		
79	0114		0 000 000 000		DUMMY				
80	0115		0 010 010 000	ALPHA9	ROM 1		GO OUTPUT CHARACTER		
81	0116		0 000 000 000		DUMMY				
82	0117		0 000 000 000		DUMMY				
83	0120		0 000 000 000		DUMMY				
84	0121		0 000 000 000		DUMMY				
85	0122		0 000 000 000		DUMMY				
86	0123		0 000 000 000		DUMMY				
87	0124		0 000 000 000		DUMMY				
88	0125		0 000 000 000		DUMMY				
89	0126		0 000 000 000		DUMMY				
90	0127		0 000 000 000		DUMMY				
91	0130		0 000 000 000		DUMMY				
92	0131		0 000 000 000		DUMMY				
93	0132		0 000 000 000		DUMMY				
94	0133		0 000 000 000		DUMMY				
95	0134		0 000 000 000		DUMMY				
96	0135		0 000 000 000		DUMMY				
97	0136		0 000 000 000		DUMMY				
98	0137		0 000 000 000		DUMMY				
99	0140		0 000 000 000		DUMMY				
100	0141		0 000 000 000		DUMMY				
101	0142		0 111 110 000	ALPHA3	CTT		LOAD IN T-REG		
102	0143		1 100 001 100		PT12		INITIALIZE POINTER FOR SHIFT		
103	0144		0 000 011 100	ROTATE	PRS		SHIFT POINTER		
104	0145		1 001 001 110		W,SRC		SHIFT DATA		
105	0146		0 000 101 100		YP0		IS SHIFT COMPLETE ?		
106	0147	0144	0 110 010 011		BRN	ROTATE	NO, CONTINUE TO SHIFT		
107	0150		0 101 000 000		IS2		YES, SHIFT I/O REG		
108	0151		0 100 101 000		IXT		SAVE 8 BITS FIRST		
109	0152		1 001 000 000		IS1		PREPARE TO SHIFT		
110	0153	0075	0 011 111 001		JSB	SLEFT	SHIFT I/O REG LEFT 8 BITS		
111	0154		0 101 000 000		IS2		PREPARE TO FINISH ROTATE		
112	0155		0 011 001 000		IOR		LOAD FIRST 8 BITS AT END OF REG		

TYPEWRITER INTERFACE LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		1 001 000 000		ISI		PREPARE TO STORE DATA
114	0157		0 111 111 000		TTC		SAVE ROTATED WORD IN C-REG
115	0160		1 011 110 000		DTDS		STORE IN DATA STORAGE
116	0161		0 101 000 000		IS2		PREPARE TO RECALL 8 BITS
117	0162		0 010 001 000		XOR		CLEAR T-REG FIRST
118	0163		0 100 101 000		IXT		RECALL FIRST 8 BITS
119	0164		1 001 000 000		ISI		PREPARE TO OUTPUT
120	0165	0115	0 100 110 111		BRN	ALPHA9	GO TO OUTPUT ROUTINE
121	0166		0 001 111 010	NOT0	XS,CM1		IS REG ADDRESS GREATER THAN 99 ?
122	0167	0333	1 101 101 111		BRN	DSERR	YES, GO TO ERROR ROUTINE
123	0170		0 001 001 100		PT1		NO, SET POINTER FOR GREATER THAN 9 TEST
124	0171		0 001 100 010		P,CM1		IS REG ADDRESS GREATER THAN 9 ?
125	0172	0333	1 101 101 111		BRN	DSERR	YES, GO TO ERROR ROUTINE
126	0173		1 110 101 110		W,AXC		LOAD ADDRESS IN C-REG
127	0174		1 011 001 100		PT11		SET POINTER TO LOAD ADDRESS
128	0175		1 110 011 000		LDC14		LOAD ADDRESS
129	0176		1 011 000 100		SS11		SET FLAG
130	0177	0311	1 100 100 111		BRN	EREG	ENABLE REG
131	0200		1 100 101 000	STO	DNR		RECALL ALPHA WORD FROM STACK
132	0201		0 000 001 100		PT0		SET POINTER TO CLEAR SHIFT DIGIT
133	0202		0 011 000 010		P,ZTC		CLEAR SHIFT DIGIT
134	0203	0326	1 101 011 101		JSB	STOREG	PREPARE TO STORE WORD IN DATA STORAGE
135	0204		0 100 010 000	SYNTAX	ROM 2		GO TO ROM 2 ERROR ROUTINE
136	0205		0 100 010 100	CPAR1	YS4		WAS LAST REG FILLED ?
137	0206	0200	1 000 000 011		BRN	STO	NO, COMPLETE LAST WORD
138	0207		0 100 100 100		RS4		YES, RESET FLAG
139	0210		0 110 101 000		STA		RECALL NEW ALPHA WORD (TO BE DESTROYED)
140	0211	0365	1 111 010 111		BRN	EXIT	CONTINUE ROUTINE
141	0212	0341	1 110 001 001	OPAR	JSB	OPLEV=7?	DOES OP LEVEL = 7 ?
142	0213		0 110 101 000		STA		YES, PREPARE STACK
143	0214		0 110 101 000		STA		RECALL REG ADDRESS TO A-REG
144	0215	0274	1 011 110 101	PREP	JSB	ADTEST	TEST ADDRESS AND ENABLE REGISTER
145	0216	0106	0 100 011 101		JSB	STOAID	STORE NEW ADDRESS IN STACK
146	0217		0 011 001 110		W,ZTC		CLEAR C-REG
147	0220		0 111 110 000		CTT		CLEAR T-REG
148	0221		0 000 001 100		PT0		SET POINTER FOR SHIFT DIGIT
149	0222		0 110 011 000		LDC6		LOAD SHIFT DIGIT
150	0223		0 100 101 000		CTS		SAVE ALPHA WORD IN STACK
151	0224		1 010 101 000		MTC		RECALL TFLAG FOR UPDATE
152	0225		1 101 001 100		PT13		SET POINTER TO UPDATE OP LEVEL
153	0226		0 110 011 000		LDC6		SET OP LEVEL TO 6
154	0227	0057	0 010 111 111		BRN	DIGEX1	RETURN TO SUPERVISOR
155	0230		0 000 001 100	STORE9	PT0		SET POINTER ON SHIFT DIGIT
156	0231		0 110 000 010		P,CTA		LOAD SHIFT DIGIT IN A-REG
157	0232		0 101 100 010	SHIFT1	P,CM1C		DECREMENT SHIFT DIGIT, WAS IT 0?
158	0233	0316	1 100 111 011		BRN	SHIFT	NO, SHIFT DATA
159	0234		0 011 000 010		P,ZTC		CLEAR THE 9 LEFT FROM SHIFT DIGIT
160	0235		0 101 000 000		IS2		YES, PREPARE TO ADD CHARACTER
161	0236		0 011 001 000		IOR		PLACE CHARACTER AFTER LAST CHARACTER
162	0237		1 001 000 000		IS1		PREPARE TO STORE WORD
163	0240		0 111 111 000		TTC		BRING WORD TO C-REG
164	0241		1 101 100 010	LASTCH?	P,AM1A		WAS THIS THE LAST CHARACTER?
165	0242	0250	1 010 100 011		BRN	STORE2	NO, SAVE ALPHA WORD
166	0243		1 011 110 000		DTDS		YES, LOAD ALPHA WORD IN D.S.
167	0244		0 100 000 100		SS4		SET REG FILLED FLAG
168	0245		0 110 101 000		STA		RECALL LAST ADDRESS TO A-REG
169	0246		1 111 101 010		X,AP1A		INCREMENT REG ADDRESS
170	0247	0215	1 000 110 111		BRN	PREP	PREPARE NEXT REG
171	0250		1 110 100 010	STORE2	P,AXC		RESTORE SHIFT DIGIT
172	0251		0 100 101 000		CTS		RESTORE ALPHA WORD IN STACK

TYPEWRITER INTERFACE LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 011 001 110	STORE3	W,STC		CLEAR C-REG
174	0253		0 111 110 000		CTT		CLEAR T-REG
175	0254	0060	0 011 000 011		BRN	DIGEX2	RETURN TO SUPERVISOR
176	0255		1 000 101 110	OPLEV=1?	W,BXC		SAVE DATA IN B-REG
177	0256		1 010 101 000		MTC		RECALL TFLAG
178	0257		0 101 111 110		S,CM1C		DOES OP LEVEL=0?
179	0260		0 101 111 110		S,CM1C		DOES OP LEVEL=1?
180	0261	0204	1 000 010 011		BRN	SYNTAX	NO, GO TO SYNTAX ERROR ROUTINE
181	0262		0 110 101 000		STA		YES, RECALL ADRESS FROM STACK
182	0263		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
183	0264		1 000 010 100	TESTPA	YS8		IN PRINT ROUTINE?
184	0265	0033	0 001 101 111		BRN	NCHAR	NO, CHECK NEXT CHARACTER
185	0266		0 110 101 000	INCREG1	STA		YES, RECALL ADDRESS TO A-REG
186	0267		1 101 100 110		M,AM1A		DECREMENT REL ADDRESS, LAST REG?
187	0270	0011	0 000 100 111		BRN	INCREG2	NO, INCREMENT REG
188	0271		1 000 100 100		RS8		YES, RESET STATUS BIT
189	0272		1 001 100 100		RS9		RESET STATUS BIT
190	0273	0060	0 011 000 011		BRN	DIGEX2	RETURN TO SUPERVISOR
191	0274		0 000 000 000		DUMMY		
192	0275		1 110 101 110	ADTEST	W,AXC		LOAD ADDRESS IN C-REG
193	0276		0 110 001 010	REG0	X,CTA		ADDRESS TO A-REG
194	0277		1 010 001 100		PT10		SET POINTER FOR SHIFT
195	0300		0 100 001 110	SHIFTR	W,SLA		SHIFT ADDRESS
196	0301		0 000 011 100		PRS		SHIFT POINTER
197	0302		0 000 101 100		YP0		IS SHIFT COMPLETE?
198	0303	0300	1 100 000 011		BRN	SHIFTR	NO, CONTINUE SHIFT
199	0304		1 111 100 010		P,AP1A		LOAD EXPONENT
200	0305		1 111 100 010		P,AP1A		PART OF ADDRESS
201	0306		0 001 111 110		S,CM1		IS TYPEWRITER REG CALLED?
202	0307	0061	0 011 000 111		BRN	TYPE	YES, FINISH ADDRESS
203	0310		1 110 101 110		W,AXC		NO, FINAL ADDRESS TO C-REG
204	0311		1 001 110 000	EREG	ATDS		ENABLE REG.
205	0312		1 011 010 100		YS11		WAS REG FOUND?
206	0313	0333	1 101 101 111		BRN	DSERR	NO, GO TO ERROR ROUTINE
207	0314		1 011 100 100		RS11		YES, RESET FLAG
208	0315		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
209	0316	0075	0 011 111 001	SHIFT	JSB	SLEFT	SHIFT LEFT 8 BITS
210	0317	0232	1 001 101 011		BRN	SHIFT1	CHECK IF SHIFT IS COMPLETE
211	0320		0 111 111 000	SRIGHT	TTC		LOAD IN C-REG FOR RIGHT SHIFT
212	0321		1 001 001 110	SR	W,SRC		SHIFT RIGHT
213	0322		0 000 011 100		PRS		SHIFT POINTER
214	0323		0 000 101 100		YP0		IS SHIFT COMPLETE?
215	0324	0321	1 101 000 111		BRN	SR	NO, CONTINUE SHIFT
216	0325		0 111 110 000		CTT		YES, RELOAD IN T-REG.
217	0326	0360	1 111 000 011		BRN	PMS7	CONTINUE ROUTINE
218	0327		1 011 110 000	STOREG	DTDS		STORE LAST ALPHA WORD IN DATA STORAGE
219	0330		1 010 101 000		MTC		RECALL TFLAG FOR UPDATE
220	0331		0 110 101 000		STA		RECALL LAST REG ADDRESS FOR DISPLAY
221	0332	0367	1 111 011 111		BRN	SOL	CONTINUE ROUTINE
222	0333		1 010 101 000	DSERR	MTC		RECALL TFLAG
223	0334		0 011 011 110		S,ZTC		SET OP LEVEL TO 0
224	0335		0 010 101 000		CXM		STORE NEW FLAG
225	0336		0 100 010 000		ROM 2		GO TO ERROR ROUTINE
226	0337		0 111 111 110	OPLEV=5?	S,CP1C		DOES OP LEVEL=6? (3)
227	0340		0 111 111 110	OPLEV=6?	S,CP1C		DOES OP LEVEL=6? (2)
228	0341		0 111 111 110		S,CP1C		DOES OP LEVEL=5?
229	0342		0 111 111 110	OPLEV=7?	S,CP1C		DOES OP LEVEL=7? (6)
230	0343		0 111 111 110		S,CP1C		DOES OP LEVEL=9?
231	0334		0 111 111 110		S,CP1C		DOES OP LEVEL=8?
232	0345		0 111 111 110		S,CP1C		DOES OP LEVEL=7? (6)

206

TYPEWRITER INTERFACE LISTING (ROM 4) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
233	0346	0204	1 000 010 011		BRN	SYNTAX	NO, GO TO SYNTAX ERROR ROUTINE	
234	0347		0 000 110 000		RETURN		YES, RETURN TO CALLING ROUTINE	
235	0350	0336	1 101 111 101	STORE	JSB	OPLEV=5?	DOES OP LEVEL = 6 ?	
236	0351		0 011 001 110		W,ZTC		YES, CLEAR C-REG FOR BIT MASK	
237	0352		0 001 001 100		PT1		SET POINTER FOR MASK	
238	0353		1 111 011 000		LDC15		LOAD MASK	
239	0354		1 111 011 000		LDC15		IN C-REG	
240	0355		0 101 000 000		IS2		SELECT ONLY	
241	0356		0 010 101 000		AND		8 LSB	
242	0357		1 001 000 000		IS1		OF T-REG	
243	0360		0 100 100 100	PMS7	RS4		RESET REG FILLED FLAG	
244	0361		1 100 101 000		DNR		RECALL ALPHA FROM STACK	
245	0362	0230	1 001 100 011		BRN	STORE9	CONTINUE ROUTINE	
246	0363	0336	1 101 111 101	CPAR	JSB	OPLEV=5?	DOES OP LEVEL = 6?	
247	0364	0205	1 000 010 111		BRN	CPAR1	YES, CONTINUE ROUTINE	
248	0365		0 110 101 000	EXIT	STA		RECALL REG FOR DISPLAY	
249	0366		1 101 101 010		X,AM1A		RESTORE PROPER VALUE OF LAST REG USED	
250	0367		0 011 011 110	SOL	S,ZTC		SET OP LEVEL TO 0	
251	0370		1 100 001 100		PT12		SET POINTER TO OP MODE	
252	0371		0 001 100 010		P,CM1		PROGRAM MODE ?	
253	0372	0057	0 010 111 111		BRN	DIGEX1	YES, DO NOT DISPLAY REG	
254	0373		0 010 010 000		ROM 1		NO, DISPLAY LAST REG	
255	0374		0 010 101 000	ALPHA2	CXM		STORE NEW FLAG	
256	0375		1 011 111 000		DSTC		RECALL CONTENTS OF DATA STORAGE	
257	0376	0142	0 110 001 011		BRN	ALPHA3	CONTINUE ROUTINE	
258	0377		0 000 000 000		DUMMY			

TYPEWRITER INTERFACE LISTING (ROM 5)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000		1 010 001 100		PT10	SET POINTER TO CONVERT 10 BITS
4	0001		0 101 000 000	LOADP	IST2	PREPARE TO LOAD PROGRAM COUNTER
5	0002		1 000 101 000		PTT	LOAD PROGRAM COUNTER IN T-REG
6	0003		1 001 000 000		IS1	PREPARE TO CONVERT PROGRAM CNTR TO BCD
7	0004	0313	1 100 110 001	PLOADED	JSB BIN-BCD	CONVERT PROGRAM COUNTER TO BCD NUMBER
8	0005		1 100 101 110		W,AXB	PROGRAM COUNTER TO A-REG
9	0006		0 100 001 110	LSHIFT	W,SLA	SHIFT A-REG LEFT
10	0007		0 000 111 100		PLS	SHIFT POINTER LEFT
11	0010		1 001 101 100		YP9	IS ADJ. COMPLETE?
12	0011	0006	0 000 011 011		BRN LSHIFT	NO, CONTINUE TO SHIFT
13	0012		0 011 011 110		S,ZTC	CLEAR SIGN DIGIT
14	0013		0 101 111 110		S,CM1C	LOAD NINE IN SIGN DIGIT
15	0014	0207	1 000 100 001		JSB PCOUT	CONTINUE ROUTINE
16	0015		1 100 010 000	CPAR	ROM 6	GO TO MNEMONIC FOR)
17	0016		0 011 001 110	T377	W,ZTC	CLEAR C-REG
18	0017		0 111 110 000		CTT	CLEAR T-REG
19	0020		0 101 000 000		IS2	PREPARE TO LOAD CODE IN T-REG
20	0021		0 100 101 000		IXT	LOAD CODE IN T-REG
21	0022		1 001 000 000		IS1	PREPARE TO TEST CODE
22	0023		1 101 001 100		PT13	SET POINTER FOR 377 TEST
23	0024		1 111 011 000		LDC15	LOAD MASK
24	0025		0 000 000 000		NOP	FOR END OF
25	0026		1 111 011 000		LDC15	OCTAL FIELD TEST
26	0027	0266	1 011 011 011		BRN TEST	GO TO TEST ROUTINE
27	0030		1 011 010 100	TEST1	YS11	WAS THERE A CARRY DURING TEST?
28	0031	0073	0 011 101 111		BRN 377	NO, CODE IS 377
29	0032		0 101 000 000		IS2	YES, BEGIN BIN-OCT CONV BY LOADING 2 ZEROES
30	0033		0 101 001 000		SRT	FIRST ZERO
31	0034		0 101 001 000		SRT	SECOND ZERO
32	0035		1 001 000 000		IS1	LOAD T-REG
33	0036		1 101 001 100		PT13	SET POINTER FOR BINARY TO OCTAL CONV.
34	0037		0 111 111 000	BIN-OCT	TTC	INTO C-REG
35	0040		0 110 000 010		P,CTA	LOAD DIGIT IN A-REG
36	0041		0 011 000 010		P,ZTC	CLEAR DIGIT IN C-REG
37	0042		0 111 110 000		CTT	RELOAD IN T-REG
38	0043		0 101 000 000		IS2	INSERT ANOTHER
39	0044		0 101 001 000		SRT	ZERO IN THE
40	0045		1 001 000 000		IS1	BINARY NUMBER
41	0046		0 000 011 100		PRS	SHIFT POINTER
42	0047		1 010 101 100		YP10	IS CONVERSION DONE?
43	0050	0037	0 001 111 111		BRN BIN-OCT	NO, CONTINUE
44	0051		1 011 001 110		W,SRA	YES, SHIFT BCD NUMBER
45	0052		0 011 001 110	OUTPUT	W,ZTC	CLEAR C-REG
46	0053		0 101 111 110		S,CM1C	SET SIGN DIGIT TO 9
47	0054		1 110 100 110	OUTPUT1	M,AXC	LOAD DATA IN C-REG
48	0055		0 110 000 110		M,CTA	RELOAD IN A-REG
49	0056	0111	0 100 101 001		JSB CTI	OUTPUT FIRST DIGIT
50	0057		0 000 111 100		PLS	SHIFT POINTER
51	0060		0 100 001 110		W,SLA	SHIFT DATA
52	0061		1 101 101 100		YP13	IS OUTPUT COMPLETE?

TYPEWRITER INTERFACE LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
53	0062	0054	0 010 110 011		BRN	OUTPUT1	NO, CONTINUE
54	0063		0 000 010 100	KEY	YS0		YES, IS KEY DOWN?
55	0064	0374	1 111 110 011		BRN	LIST2	NO, CONTINUE TO LIST
56	0065		0 000 100 100		RS0		YES, RESET STATUS BIT
57	0066	0102	0 100 001 101	RETURN	JSB	CR/LF	DO A CR/LF
58	0067		1 100 101 000		DNR		RECALL DATA FROM STACK
59	0070		1 000 101 110		W,BXC		SAVE DATA IN B-REG
60	0071		1 100 101 000		DNR		PREPARE STACK FOR STORE
61	0072		0 100 010 000		ROM 2		RETURN TO SUPERVISOR VIA ROM 2
62	0073		0 111 100 100	377	RS7		RESET QUOTE FIELD FLAG
63	0074		1 011 001 100		PT11		SET POINTER FOR MNEMONIC GENERATION
64	0075	0015	0 000 110 111		BRN	CPAR	GO TO ")"
65	0076		0 101 000 000	TAB	IS2		PREPARE TO OUTPUT TAB
66	0077	0037	0 010 000 001		LIO	100	LOAD CODE FOR TAB
67	0100		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
68	0101		1 001 000 000		IS1		PREPARE TO OUTPUT
69	0102	0351	1 110 100 111		BRN	WAIT	OUTPUT TAB
70	0103		0 101 000 000	CR/LF	IS2		PREPARE TO OUTPUT CR/LF
71	0104	0002	0 000 001 101		LIO	006	LOAD CODE FOR CR/LF
72	0105		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
73	0106		1 001 000 000		IS1		PREPARE TO OUTPUT
74	0107	0351	1 110 100 111		BRN	WAIT	OUTPUT CR/LF
75	0110		1 011 001 100	MNEMONIC	PT11		SET POINTER TO BEGIN MNEMONIC GENERATION
76	0111		1 100 010 000		ROM 6		GO TO ROM 6 TO GENERATE MNEMONICS
77	0112		0 111 110 000	CTI	CTT		LOAD DATA IN T-REG
78	0113		0 101 000 000		IS2		PREPARE TO LOAD DATA IN I/O-REG
79	0114		0 100 101 000		IXT		LOAD DATA IN I/O-REG
80	0115		1 001 000 000		IS1		PREPARE TO OUTPUT
81	0116	0351	1 110 100 111		BRN	WAIT	OUTPUT DATA
82	0117		0 101 010 100		YS5		WAS LAST INSTR A LBL?
83	0120	0144	0 110 010 011		BRN	JMP/SUB	NO, TEST FOR JMP/SUB
84	0121	0240	1 010 000 101	LBL1	JSB	STACK2	YES, READ NEXT 2 INSTRUCTIONS & STACK IN C-REG
85	0122		0 101 100 100		RS5		RESET LBL FLAG
86	0123	0126	0 101 011 011		BRN	LBL	CONTINUE ROUTINE
87	0124	0350	1 110 100 101		JSB	WAIT	OUTPUT MNEMONIC
88	0125		1 110 010 000		ROM 7		RETURN TO SHIFT MNEMONIC
89	0126		0 110 001 110	LBL	W,CTA		LOAD COMPLETED LABEL IN A-REG
90	0127		1 011 001 110		W,SRA		SHIFT LBL.
91	0130		1 010 001 100		PT10		SET POINTER FOR LAST DIGIT OUT
92	0131		0 110 010 100		YS6		WAS LAST INSTR SUB OR JMP?
93	0132	0136	0 101 111 011		BRN	CR	NO, DO A CR
94	0133		0 110 100 100		RS6		YES, RESET FLAG
95	0134	0075	0 011 111 001	LBL3	JSB	TAB	TAB FOR LBL POSITION
96	0135	0052	0 010 101 011		BRN	OUTPUT	OUTPUT LBL
97	0136		0 101 000 000	CR	IS2		LOAD CODE
98	0137	0000	0 000 000 101		LIO	002	FOR A CARRIAGE
99	0140		0 101 101 000		SRI		RETURN AND
100	0141		1 001 000 000		IS1		THEN OUTPUT
101	0142	0350	1 110 100 101		JSB	WAIT	THAT CODE
102	0143	0134	0 101 110 011		BRN	LBL3	NOW TAB AND OUTPUT LBL.
103	0144		0 110 010 100	JMP/SUB	YS6		WAS INSTR JMP OR SUB?
104	0145	0204	1 000 010 011		BRN	LEND	NO, TEST FOR END
105	0146	0240	1 010 000 101	JMP1	JSB	STACK2	YES, READ NEXT 2 INSTRUCTIONS & STACK IN C-REG
106	0147		0 110 100 100		RS6		RESET REL/ABS FLAG
107	0150		1 000 010 100		YS8		WAS INSTR SUBC?
108	0151	0154	0 110 110 011		BRN	JMP4	NO, OUTPUT ABS/REL ADDRESS
109	0152		1 000 100 100		RS8		YES, RESET FLAG
110	0153	0063	0 011 001 111		BRN	KEY	TEST FOR KEY DOWN
111	0154		1 001 001 110	JMP4	W,SRC		SHIFT ADDRESS TO
112	0155		0 000 111 100		PLS		LOW END OF WORD

TYPEWRITER INTERFACE LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		1 010 101 100		YP10		IS SHIFT DONE?
114	0157	0154	0 110 110 011		BRN	JMP4	NO, CONTINUE
115	0160		0 111 110 000		CTT		YES, RELOAD IN T-REG
116	0161		0 111 010 100		YS7		IS JMP- FLAG SET?
117	0162	0171	0 111 100 111		BRN	CONVERT	NO, CONVERT ADDRESS TO BCD NUMBER
118	0163		0 111 100 100		RS7		YES, RESET FLAG
119	0164		0 101 000 000		IS2		PREPARE TO COMPLEMENT ADDRESS
120	0165		0 010 001 000		XOR		CLEAR T-REG
121	0166		1 101 001 000		TDEC		SET T-REG TO ALL ONES
122	0167		0 010 001 000		XOR		COMPLEMENT C-REG AND LOAD RESULT IN T-REG
123	0170		1 101 001 000		TDEC		ADJUST ADDRESS FOR LATE TINC
124	0171		0 101 000 000	CONVERT	IS2		PREPARE TO MAKE FINAL ADJUSTMENT ON ADDRESS
125	0172		1 100 001 000		TINC		ADJUST ADDRESS
126	0173		1 001 000 000		IS1		PREPARE TO CONVERT NUMBER TO BCD
127	0174	0313	1 100 110 001		JSB	BIN-BCD	CONVERT TO BCD
128	0175		1 100 101 110	JMP5	W,AXB		LOAD ADDRESS IN A-REG
129	0176		0 000 001 100		PT0		FOR SHIFT
130	0177		0 100 001 110	JMP6	W,SLA		SHIFT ADDRESS
131	0200		0 000 111 100		PLS		SHIFT POINTER
132	0201		1 001 101 100		YP9		IS SHIFT DONE?
133	0202	0177	0 111 111 111		BRN	JMP6	NO, CONTINUE
134	0203	0052	0 010 101 011		BRN	OUTPUT	YES, OUTPUT ADDRESS
135	0204		1 000 010 100	LEND	YS8		WAS INSTR END?
136	0205	0063	0 011 001 111		BRN	KEY	NO, TEST FOR KEY DOWN
137	0206		1 000 100 100		RS8		YES, RESET FLAG
138	0207	0066	0 011 011 011		BRN	RETURN	RETURN TO SUPERVISOR
139	0210		1 110 100 110	PCOUT	M,AXC		LOAD DIGIT IN C-REG
140	0211		0 110 000 110		M,CTA		RESTORE DATA IN A-REG
141	0212	0111	0 100 101 001	BCTI	JSB	CTI	LOAD DATA IN I/O-REG AND OUTPUT
142	0213		0 100 001 110	PCOUT1	W,SLA		SHIFT DATA LEFT ONCE
143	0214		0 000 111 100		PLS		SHIFT POINTER
144	0215		1 101 101 100		YP13		IS OUTPUT COMPLETE?
145	0216	0210	1 000 100 011		BRN	PCOUT	NO, CONTINUE
146	0217	0075	0 011 111 001		JSB	TAB	YES, TAB ONCE
147	0220	0075	0 011 111 001		JSB	TAB	TAB AGAIN
148	0221		0 101 000 000	READ	IS2		PREPARE TO READ NEXT INSTRUCTION
149	0222		1 001 001 000		PDEC		DECREMENT PROGRAM COUNTER
150	0223		0 111 000 000		PINC		INCREMENT PROGRAM COUNTER AND LOAD ADDRESS
151	0224		1 001 000 000		IS1		PREPARE TO READ
152	0225		1 000 100 000		READ		READ NEXT PROGRAM INSTRUCTION
153	0226		1 001 010 100	I/O	YS9		INSIDE I/O CALL ?
154	0227	0110	0 100 100 011		BRN	MNEMONIC	NO, GENERATE MNEMONIC
155	0230	0232	1 001 101 011		BRN	YS7	YES, TEST FOR INSIDE "("
156	0231		0 100 010 000	LERROR1	ROM2		GO TO ROM 2 FOR ERROR ROUTINE
157	0232		0 111 010 100	YS7	YS7		INSIDE "(" ?
158	0233	0110	0 100 100 011		BRN	MNEMONIC	NO, GENERATE MNEMONIC
159	0234	0016	0 000 111 011		BRN	T377	YES, CONTINUE ROUTINE
160	0235		1 110 010 000	CR/LF3	ROM 7		RETURN TO ROM 7 TO CHECK TAB FLAG
161	0236	0350	1 110 100 101		JSB	WAIT	OUTPUT CR/LF
162	0237	0235	1 001 110 111		BRN	CR/LF3	CONTINUE ROTINE
163	0240		0 000 000 000		DUMMY		
164	0241		0 111 000 000	STACK2	PINC		INCREMENT PROGRAM COUNTER
165	0242		1 000 100 000		READ		READ NEXT INSTRUCTION
166	0243		0 111 001 100		PT7		SET POINTER FOR SHIFT
167	0244		0 101 000 000		IS2		PREPARE TO LOAD DATA IN T-REG
168	0245		0 100 101 000		IXT		LOAD DATA IN T-REG
169	0246		1 001 000 000		IS1		PREPARE TO READ SECOND INSTRUCTION
170	0247		0 111 000 000		PINC		INCREMENT PROGRAM COUNTER
171	0250		1 000 100 000		READ		READ SECOND INSTRUCTION
172	0251		0 101 000 000		IS2		PREPARE TO LOAD IN T-REG

TYPEWRITER INTERFACE LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 100 101 000		IXT		FIRST TO I/O, SECOND TO T-REG
174	0253		0 101 000 000	SHIFT2ND	IS2		PREPARE TO SHIFT SECOND INSTRUCTION
175	0254		0 101 001 000		SRT		SHIFT DATA
176	0255		1 001 000 000		IS1		PREPARE TO CHECK IF SHIFT DONE
177	0256		0 000 011 100		PRS		SHIFT POINTER
178	0257		1 111 101 100		YP15		IS SHIFT DONE?
179	0260	0253	1 010 101 111		BRN	SHIFT2ND	NO, CONTINUE TO SHIFT
180	0261		0 101 000 000		IS2		YES, PREPARE TO LOAD FIRST INSTRUCTION
181	0262		0 100 101 000		IXT		LOAD FIRST INSTRUCTION IN FRONT OF SECOND
182	0263		1 001 000 000		IS1		PREPARE TO BRING RESULT TO C-REG
183	0264		0 111 111 000		TTC		BRING RESULT TO C-REG
184	0265		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
185	0266		0 110 010 000	TEST	ROM 3		GO TO ROM 3 TO TEST DATA
186	0267		0 000 000 000		DUMMY		
187	0270		0 000 000 000		DUMMY		
188	0271	0030	0 001 100 011		BRN	TEST1	CHECK TEST RESULTS
189	0272		0 000 000 000		DUMMY		
190	0273		0 000 000 000		DUMMY		
191	0274		0 000 000 000		DUMMY		
192	0275		0 000 000 000		DUMMY		
193	0276		0 000 000 000		DUMMY		
194	0277		0 000 000 000		DUMMY		
195	0300		0 000 000 000		DUMMY		
196	0301		0 000 000 000		DUMMY		
197	0302		0 000 000 000		DUMMY		
198	0303		0 000 000 000		DUMMY		
199	0304		0 000 000 000		DUMMY		
200	0305		0 000 000 000		DUMMY		
201	0306		0 000 000 000		DUMMY		
202	0307		0 101 000 000	OUTPUTL	IS2		PREPARE TO OUTPUT DATA
203	0310	0017	0 001 000 001		LIO	040	LOAD I/O SELECT CODE
204	0311		0 110 100 000		GIOE		OUTPUT DATA
205	0312		1 001 000 000		IS1		PREPARE TO RETURN TO CALLING ROUTINE
206	0313		0 000 110 000		RETURN		RETURN TO CALLING ROUTINE
207	0314		0 000 101 110	BIN-BCD	W,ZTB		CLEAR B-REG SO SUM=0
208	0315		1 011 101 110		W,ZTA		CLEAR A-REG
209	0316		1 111 101 110		W,AP1A		INITIALIZE A-REG TO 1 FOR LSB
210	0317		0 111 111 000	BCD1	TTC		LOAD BINARY NUMBER IN C-REG
211	0320		0 100 101 000		CTS		STORE BINARY NUMBER IN STACK
212	0321		0 101 000 000		IS2		SHIFT BINARY
213	0322		0 101 001 000		SRT		NUMBER IN T-REG
214	0323		0 100 001 000		SLT		TO SET LSB TO 0
215	0324		0 010 001 000		XOR		COMPARE T & C-REG
216	0325		1 001 000 000		IS1		PREPARE TO CHECK RESULTS
217	0326		0 111 111 000		TTC		LOAD COMPARISON RESULT IN C-REG
218	0327		0 001 101 110		W,CM1		WAS THAT BIT A ONE?
219	0330	0333	1 101 101 111		BRN	BCD2	YES, ADD BINARY VALUE IN A TO SUM IN B
220	0331		0 010 001 110		W,BTC		NO, LOAD PREVIOUS SUM IN B-REG
221	0332	0335	1 101 110 111		BRN	BCD3	DOUBLE PRESENT BINARY NUMBER
222	0333		0 010 001 110	BCD2	W,BTC		LOAD PREVIOUS SUM IN C-REG
223	0334		0 111 001 110		W,APCC		ADD PREVIOUS SUM AND PRESENT BINARY VALUE
224	0335		0 100 101 110	BCD3	W,ATB		LOAD PRESENT BINARY VALUE IN B-REG
225	0336		1 110 001 110		W,APBA		DOUBLE BINARY VALUE IN A-REG
226	0337		1 000 101 110		W,BXC		SAVE LATEST BCD SUM IN B-REG
227	0340		1 100 101 000		DNR		RECALL BINARY NUMBER FROM STACK
228	0341		0 111 110 000		CTT		LOAD BINARY NUMBER IN T-REG
229	0342		0 101 000 000		IS2		SHIFT BINARY
230	0343		0 101 001 000		SRT		NUMBER TO
231	0344		1 001 000 000		IS1		EXAMINE NEXT BIT
232	0345		0 000 011 100		PRS		SHIFT POINTER

TYPEWRITER INTERFACE LISTING (ROM 5) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
233	0346		0 000 101 100		YP0		IS CONVERSION COMPLETE?
234	0347	0317	1 100 111 111		BRN	BCD1	NO, CONTINUE ROUTINE
235	0350		0 000 110 000		RETURN		YES, RETURN TO CALLING ROUTINE
236	0351		1 011 101 010	WAIT	X,ZTA		INITIALIZE COUNTER
237	0352		0 101 000 000	TESTFLG	IS2		PREPARE TO TEST FLAG
238	0353		0 000 100 000		YINTF		INTERROGATE FLAG STATUS BIT
239	0354		1 001 000 000		IS1		PREPARE TO CHECK FLAG STATUS
240	0355		1 011 010 100		YS11		IS PERIPHERAL READY ?
241	0356	0307	1 100 011 111		BRN	OUTPUTL	GO TO OUTPUT DATA ROUTINE
242	0357		1 011 100 100		RS11		NO, RESET STATUS BIT AND CHECK FLAG AGAIN
243	0360		1 111 101 010		X,AP1A		INCREMENT COUNTER, WAS THERE A CARRY ?
244	0361	0352	1 110 101 011		BRN	TESTFLG	NO, TEST FLAG AGAIN
245	0362		0 100 010 000	FAILURE	ROM 2		YES, GO TO ROM 2 TO PRINT ERROR MESSAGE
246	0363		0 101 000 000	LISTE	IS2		PREPARE TO CHECK IF MEMORY IS PROTECTED
247	0364	0077	0 100 000 001		LIO	200	LOAD TEST CODE
248	0365		1 110 100 000		TG8		TEST MEMORY
249	0366		1 001 000 000		IS1		PREPARE TO CHECK TEST RESULTS
250	0367		1 011 010 100		YS11		IS MEMORY PROTECTED?
251	0370	0231	1 001 100 111		BRN	LERROR1	YES, MEMORY IS PROTECTED, GO TO ERROR ROUTINE
252	0371		1 011 100 100		RS11		NO, RESET STATUS BIT
253	0372		1 000 101 110		W,BXC		RESTORE DATA TO C-REG
254	0373		0 100 101 000		CTS		SAVE DATA IN STACK
255	0374	0102	0 100 001 101	LIST2	JSB	CR/LF	OUTPUT CR/LF
256	0375		0 111 000 000		PINC		INCREMENT PROGRAM COUNTER
257	0376		0 011 001 110		W,ZTC		CLEAR C-REG
258	0377		0 111 110 000		CTT		CLEAR T-REG

TYPEWRITER INTERFACE LISTING (ROM 6)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern					
3	0000		0 111 000 100	COMPFLG	SS7			
4	0001	0050	0 010 100 011		BRN	—		
5	0002	0166	0 111 011 101	BSUBX	JSB	SSUB		
6	0003	0165	0 111 011 001		JSB	SLBL		
7	0004	0134	0 101 110 011		BRN	BLBLX		
8	0005		0 101 011 000	B*	LDC5			
9	0006		1 010 011 000	L10D	LDC10			
10	0007	0111	0 100 100 111		BRN	DONE		
11	0010	0153	0 110 110 001	B/	JSB	S/		
12	0011	0111	0 100 100 111		BRN	DONE		
13	0012	0010	0 000 100 011	/	BRN	B/	/	
14	0013	0005	0 000 010 111	AST	BRN	B*	*	
15	0014	0154	0 110 110 101	STOX	JSB	SSTO	STOX	
16	0015	0320	1 101 000 011		BRN	BX		
17	0016		0 101 011 000	)	LDC5		)	
18	0017	0105	0 100 010 111		BRN	L9D		
19	0020		0 101 011 000	=	LDC5		=	
20	0021	0103	0 100 001 111		BRN	L13D		
21	0022	0145	0 110 010 111	.	BRN	B.	.	
22	0023		1 001 010 010	0	WP,SRC		0	
23	0024		1 001 011 000		LDC9			
24	0025	0265	1 011 010 111		BRN	L0D		
25	0026		1 001 010 100	(	YS9		(	
26	0027	0142	0 110 001 011		BRN	B(		
27	0030		0 111 000 100		SS7			
28	0031	0142	0 110 001 011		BRN	B(		
29	0032		0 111 100 010	3	P,CP1C		3	
30	0033		0 111 100 010	2	P,CP1C		2	
31	0034		0 111 100 010	1	P,CP1C		1	
32	0035	0023	0 001 001 111		BRN	0		
33	0036		0 101 011 000	%	LDC5		%	
34	0037	0101	0 100 000 111		BRN	L5D		
35	0040		0 101 011 000	+	LDC5		+	
36	0041	0061	0 011 000 111		BRN	L11D		
37	0042		0 111 100 010	6	P,CP1C		6	
38	0043		0 111 100 010	5	P,CP1C		5	
39	0044		0 111 100 010	4	P,CP1C		4	
40	0045	0032	0 001 101 011		BRN	3		
41	0046	0163	0 111 010 001	CRCL	JSB	SC	CRCL	
42	0047	0370	1 111 100 011		BRN	SRCL		
43	0050		1 001 011 000	–	LDC9		–	
44	0051	0103	0 100 001 111		BRN	L13D		
45	0052		0 111 100 010	9	P,CP1C		9	
46	0053		0 111 100 010	8	P,CP1C		8	
47	0054		0 111 100 010	7	P,CP1C		7	
48	0055	0042	0 010 001 011		BRN	6		
49	0056	0163	0 111 010 001	CSTO	JSB	SC	CSTO	
50	0057	0154	0 110 110 101		JSB	SSTO		
51	0060	0111	0 100 100 111		BRN	DONE		
52	0061		1 011 011 000	L11D	LDC11			

TYPEWRITER INTERFACE LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern							
53	0062	0111	0 100 100 111		BRN	DONE				
54	0063		1 001 100 100	LE	RS9		LE			
55	0064	0156	0 110 111 101	LE1	JSB	SLE				
56	0065	0111	0 100 100 111		BRN	DONE				
57	0066		0 110 100 100	JLBLX	RS6					
58	0067		0 001 001 100	BJS	PT1					
59	0070	0162	0 111 001 101		JSB	SX				
60	0071	0351	1 110 100 111		BRN	RS5				
61	0072	0213	1 000 101 111	RND	BRN	BRND	RND			
62	0073	0160	0 111 000 101	CLR	JSB	SCL	CLR			
63	0074		0 111 011 000		LDC7					
64	0075	0136	0 101 111 011		BRN	L2D				
65	0076	0161	0 111 001 001	PRTX	JSB	SPRT	PRTX			
66	0077	0162	0 111 001 101		JSB	SX				
67	0100	0111	0 100 100 111		BRN	DONE				
68	0101		0 101 011 000	L5D	LDC5					
69	0102	0111	0 100 100 111		BRN	DONE				
70	0103		1 101 011 000	L13D	LDC13					
71	0104	0111	0 100 100 111		BRN	DONE				
72	0105		1 001 011 000	L9D	LDC9					
73	0106	0111	0 100 100 111		BRN	DONE				
74	0107		1 110 010 000	BACC	ROM 7					
75	0110		0 000 000 000		DUMMY					
76	0111		1 110 010 000	DONE	ROM 7					
77	0112		0 011 001 110		W,ZTC					
78	0113		0 001 010 000		EERA					
79	0114	0161	0 111 001 001	BPRTDS	JSB	SPRT				
80	0115	0171	0 111 101 001		JSB	SDS				
81	0116	0111	0 100 100 111		BRN	DONE				
82	0117		0 000 000 000		DUMMY					
83	0120		0 000 000 000		DUMMY					
84	0121		0 000 000 000		DUMMY					
85	0122		0 000 000 000		DUMMY					
86	0123		0 000 000 000		DUMMY					
87	0124		0 000 000 000		DUMMY					
88	0125		0 110 011 000	L6D	LDC6					
89	0126	0111	0 100 100 111		BRN	DONE				
90	0127		0 110 011 000	BH	LDC6					
91	0130	0143	0 110 001 111		BRN	L8D				
92	0131	0163	0 111 010 001	CHS	JSB	SC	CHS			
93	0132		1 110 010 000		ROM 7					
94	0133		0 000 110 000		RETURN					
95	0134		1 000 000 100	BLBLX	SS8					
96	0135	0067	0 011 011 111		BRN	BJS				
97	0136		0 010 011 000	L2D	LDC2					
98	0137	0111	0 100 100 111		BRN	DONE				
99	0140	0161	0 111 001 001	PRTACC	JSB	SPRT	PRTACC			
100	0141	0107	0 100 011 111		BRN	BACC				
101	0142		0 101 011 000	B(	LDC5					
102	0143		1 000 011 000	L8D	LDC8					
103	0144	0111	0 100 100 111		BRN	DONE				
104	0145		0 001 011 000	B.	LDC1					
105	0146		1 110 011 000	L14D	LDC14					
106	0147	0111	0 100 100 111		BRN	DONE				
107	0150	0164	0 111 010 101	JMP–	JSB	SJMP	JMP–XXXX	WHERE XXXX = NUMBER OF STEPS		
108	0151	0000	0 000 000 011		BRN	COMPFLG				
109	0152	0164	0 111 010 101	JMP+	JSB	SJMP	JMP+XXXX	WHERE XXXX = NUMBER OF STEPS		
110	0153	0040	0 010 000 011		BRN	+				
111	0154		1 110 010 000	S/	ROM 7					
112	0155		1 110 010 000	SSTO	ROM 7					

TYPEWRITER INTERFACE LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
113	0156		1 110 010 000	SD	ROM 7		
114	0157		1 110 010 000	SLE	ROM 7		
115	0160		1 110 010 000	SRND	ROM 7		
116	0161		1 110 010 000	SCL	ROM 7		
117	0162		1 110 010 000	SPRT	ROM 7		
118	0163		1 110 010 000	SX	ROM 7		
119	0164		1 110 010 000	SC	ROM 7		
120	0165		1 110 010 000	SJMP	ROM 7		
121	0166		1 110 010 000	SLBL	ROM 7		
122	0167		1 110 010 000	SSUB	ROM 7		
123	0170		1 110 010 000	SEX	ROM 7		
124	0171		1 110 010 000	SIF	ROM 7		
125	0172		1 110 010 000	SDS	ROM 7		
126	0173		1 110 010 000	SFLG	ROM 7		
127	0174		0 010 011 000	BLNX	LDC2		
128	0175		1 100 011 000		LDC12		
129	0176		0 010 011 000		LDC2		
130	0177		1 110 011 000	BNX	LDC14		
131	0200	0320	1 101 000 011		BRN	BX	
132	0201	0162	0 111 001 101	BX/12	JSB	SX	
133	0202	0153	0 110 110 001		JSB	S/	
134	0203		1 001 011 000		LDC9		
135	0204		0 001 011 000		LDC1		
136	0205	0033	0 001 101 111		BRN	2	
137	0206	0166	0 111 011 101	BSUBL	JSB	SSUB	
138	0207	0165	0 111 011 001	BLBL	JSB	SLBL	LBL
139	0210	0111	0 100 100 111		BRN	DONE	
140	0211		0 000 000 000		DUMMY		
141	0212		0 000 000 000		DUMMY		
142	0213	0157	0 111 000 001	BRND	JSB	SRND	
143	0214	0111	0 100 100 111		BRN	DONE	
144	0215		0 110 011 000	BEND	LDC6		
145	0216		0 101 011 000		LDC5		
146	0217		0 110 011 000	BND	LDC6		
147	0220		1 110 011 000	BD	LDC14		
148	0221	0155	0 110 111 001		JSB	SD	
149	0222		1 000 000 100	SS8	SS8		
150	0223	0111	0 100 100 111		BRN	DONE	
151	0224	0002	0 000 001 011	SUBLBX	BRN	BSUBX	SUBLBX
152	0225	0206	1 000 011 011	SUBLBL	BRN	BSUBL	SUBLBL
153	0226	0166	0 111 011 101	SUB	JSB	SSUB	SUBXXXX WHERE XXXX = LOCATION OF SUBROUTINE
154	0227	0111	0 100 100 111		BRN	DONE	
155	0230		1 001 011 000	BI/X	LDC9		
156	0231		0 001 011 000		LDC1		
157	0232	0153	0 110 110 001		JSB	S/	
158	0233	0162	0 111 001 101		JSB	SX	
159	0234	0111	0 100 100 111		BRN	DONE	
160	0235	0154	0 110 110 101	BSTO	JSB	SSTO	
161	0236	0111	0 100 100 111		BRN	DONE	
162	0237		0 000 000 000		DUMMY		
163	0240		0 111 101 110		W,CP1C		F01
164	0241		0 111 101 110		W,CP1C		F02
165	0242		0 111 101 110		W,CP1C		F03
166	0243		0 111 101 110		W,CP1C		F04
167	0244		0 111 101 110		W,CP1C		F05
168	0245		0 111 101 110		W,CP1C		F06
169	0246		0 111 101 110		W,CP1C		F07
170	0247		0 111 101 110		W,CP1C		F08
171	0250		0 111 101 110		W,CP1C		F09
172	0251		0 111 101 110		W,CP1C		F10

TYPEWRITER INTERFACE LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
173	0252		0 111 101 110		W,CP1C	F11
174	0253		0 111 101 110		W,CP1C	F12
175	0254		0 111 101 110		W,CP1C	F13
176	0255		0 111 101 110		W,CP1C	F14
177	0256		0 111 101 110		W,CP1C	F15
178	0257		1 110 010 000		ROM 7	
179	0260	0160	0 111 000 101	BCLDS	JSB SCL	
180	0261	0171	0 111 101 001		JSB SDS	
181	0262	0111	0 100 100 111		BRN DONE	
182	0263	0154	0 110 110 101	STOP	JSB SSTO	STOP
183	0264		0 111 011 000		LDC7	
184	0265		0 000 011 100	LØD	PRS	
185	0266	0111	0 100 100 111		BRN DONE	
186	0267	0230	1 001 100 011	I/X	BRN B1/X	I/X
187	0270	0167	0 111 100 001	EXPX	JSB SEX	EXPX
188	0271		0 111 011 000		LDC7	
189	0272		0 000 011 000		LDCØ	
190	0273	0162	0 111 001 101		JSB SX	
191	0274	0111	0 100 100 111		BRN DONE	
192	0275	0170	0 111 100 101	IFØ	JSB SIF	IFØ
193	0276	0023	0 001 001 111		BRN Ø	
194	0277	0171	0 111 101 001	BDS+	JSB SDS	
195	0300	0040	0 010 000 011		BRN +	
196	0301	0171	0 111 101 001	BDS–	JSB SDS	
197	0302	0050	0 010 100 011		BRN –	
198	0303		0 111 011 000	BSFLG	LDC7	
199	0304		0 011 011 000		LDC3	
200	0305	0172	0 111 101 101	BFLG	JSB SFLG	
201	0306	0111	0 100 100 111		BRN DONE	
202	0307	0164	0 111 010 101	BJMPX	JSB SJMP	
203	0310	0165	0 111 011 001		JSB SLBL	
204	0311	0066	0 011 011 011		BRN JLBLX	
205	0312	0170	0 111 100 101	IF+	JSB SIF	IF+
206	0313	0040	0 010 000 011		BRN +	
207	0314	0160	0 111 000 101	BCL	JSB SCL	
208	0315	0111	0 100 100 111		BRN DONE	
209	0316	0170	0 111 100 101	IF–	JSB SIF	IF–
210	0317	0050	0 010 100 011		BRN –	
211	0320	0162	0 111 001 101	BX	JSB SX	
212	0321	0111	0 100 100 111		BRN DONE	
213	0322	0170	0 111 100 101	IFLE	JSB SIF	IFLE
214	0323	0064	0 011 010 011		BRN LE1	
215	0324		0 000 000 000		DUMMY	
216	0325	0114	0 100 110 011	PRTDS	BRN BPRTDS	PRTDS
217	0326	0260	1 011 000 011	CLDS	BRN BCLDS	CLDS
218	0327	0277	1 011 111 111	DS+	BRN BDS+	DS+
219	0330	0301	1 100 000 111	DS–	BRN BDS–	DS–
220	0331	0235	1 001 110 111	STO	BRN BSTO	STO
221	0332		1 110 010 000	ACC+	ROM 7	ACC+
222	0333		1 110 010 000	ACC–	ROM 7	ACC–
223	0334		1 110 010 000	ACC*	ROM 7	ACC*
224	0335		1 110 010 000	ACC/	ROM 7	ACC/
225	0336	0370	1 111 100 011	RCL	BRN SRCL	RCL
226	0337	0167	0 111 100 001	EXCH	JSB SEX	EXCH
227	0340	0163	0 111 010 001		JSB SC	
228	0341	0127	0 101 011 111		BRN BH	
229	0342		1 110 010 000	RET	ROM 7	RET
230	0343	0164	0 111 010 101	BJMPL	JSB SJMP	
231	0344	0207	1 000 011 111		BRN BLBL	
232	0345		0 000 000 000		DUMMY	

TYPEWRITER INTERFACE LISTING (ROM 6) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern						
233	0346	0303	1 100 001 111	SAFLG	BRN	BSFLG	SFLG		
234	0347	0170	0 111 100 101	IFFLG	JSB	SIF	IFFLG		
235	0350	0305	1 100 010 111		BRN	BFLG			
236	0351		0 101 100 100	RS5	RS5				
237	0352	0111	0 100 100 111		BRN	DONE			
238	0353	0307	1 100 011 111	JMPLBX	BRN	BJMPX	JMPLBX		
239	0354	0343	1 110 001 111	JMPLBL	BRN	BJMPL	JMPLBL		
240	0355	0164	0 111 010 101	JMP	JSB	SJMP	JMPXXXX	WHERE XXXX = LOCATION OF JUMP	
241	0356	0111	0 100 100 111		BRN	DONE			
242	0357		0 000 000 000		DUMMY				
243	0360		0 000 000 000		DUMMY				
244	0361		0 000 000 000		DUMMY				
245	0362	0201	1 000 000 111	X/12	BRN	BX/12	X/12		
246	0363		0 110 011 000	ADV	LDC6		ADV		
247	0364		0 001 011 000		LDC1				
248	0365	0155	0 110 111 001		JSB	SD			
249	0366		0 111 011 000		LDC7				
250	0367	0125	0 101 010 111		BRN	L6D			
251	0370		0 111 011 000	SRCL	LDC7				
252	0371		0 010 011 000		LDC2				
253	0372	0314	1 100 110 011		BRN	BCL			
254	0373	0215	1 000 110 111	SAEND	BRN	BEND	END		
255	0374		1 110 010 000	LOGX	ROM 7		LOGX		
256	0375	0174	0 111 110 011	LNX	BRN	BLNX	LNX		
257	0376		1 110 010 000	CALL	ROM 7		CALL		
258	0377		0 000 000 000		DUMMY				

TYPEWRITER INTERFACE LISTING (ROM 7)

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
3	0000		0 110 011 000		LDC6	
4	0001		0 001 011 000		LDC1	
5	0002		0 110 011 000		LDC6	
6	0003		1 100 011 000		LDC12	
7	0004		0 110 011 000		LDC6	
8	0005		1 100 011 000		LDC12	
9	0006		1 001 000 100	CALL1	SS9	
10	0007	0112	0 100 101 011		BRN DONE	
11	0010		0 000 000 000		DUMMY	
12	0011		1 100 010 000	/	ROM 6	
13	0012		1 100 010 000	AST	ROM 6	
14	0013		0 010 101 000	FLG1	CXM	RESTORE FLAG
15	0014		0 100 010 000		ROM 2	CONTINUE TESTS IN ROM 2
16	0015		0 010 101 000		CXM	EXCHANGE FLAG AND DATA
17	0016		0 001 111 110		S,CM1	DOES OP LEVEL = 0?
18	0017	0013	0 000 101 111		BRN FLG1	NO, CONTINUE ROUTINE
19	0020		1 000 001 100		PT8	YES, SET POINTER TO LOAD FLAG
20	0021	0324	1 101 010 011		BRN SETFLG	CONTINUE ROUTINE
21	0022		0 111 011 000	BSTO	LDC7	
22	0023		0 011 011 000		LDC3	
23	0024		0 111 011 000		LDC7	
24	0025		0 100 011 000		LDC4	
25	0026		0 110 011 000		LDC6	
26	0027	0071	0 011 100 111		BRN L15R	
27	0030		0 111 011 000	BRND	LDC7	
28	0031		0 010 011 000		LDC2	
29	0032		0 110 011 000		LDC6	
30	0033		1 110 011 000		LDC14	
31	0034	0063	0 011 001 111		BRN BD	
32	0035		0 000 000 000		DUMMY	
33	0036		0 000 000 000		DUMMY	
34	0037		1 100 010 000	+	ROM 6	
35	0040		0 110 011 000	BCL	LDC6	
36	0041		0 011 011 000		LDC3	
37	0042		0 110 011 000		LDC6	
38	0043		1 100 011 000		LDC12	
39	0044	0132	0 101 101 011		BRN RETURN	
40	0045		0 000 000 000		DUMMY	
41	0046		0 000 000 000		DUMMY	
42	0047		1 100 010 000	–	ROM 6	
43	0050		0 111 011 000	BPRT	LDC7	
44	0051		0 000 011 000		LDC0	
45	0052		0 111 011 000		LDC7	
46	0053		0 010 011 000		LDC2	
47	0054		0 111 011 000		LDC7	
48	0055	0064	0 011 010 011		BRN L4R	
49	0056		0 110 011 000	BLE	LDC6	
50	0057		1 100 011 000		LDC12	
51	0060		0 110 011 000		LDC6	
52	0061		0 101 011 000		LDC5	

TYPEWRITER INTERFACE LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
53	0062	0132	0 101 101 011		BRN	RETURN
54	0063		0 110 011 000	BD	LDC6	
55	0064		0 100 011 000	L4R	LDC4	
56	0065	0132	0 101 101 011		BRN	RETURN
57	0066		0 000 000 000		DUMMY	
58	0067		0 000 010 000	SUPVR	ROM 0	RETURN TO SUPERVISOR IN ROM 0
59	0070		0 001 011 000	B/	LDC1	
60	0071		1 111 011 000	L15R	LDC15	
61	0072	0132	0 101 101 011		BRN	RETURN
62	0073		0 110 011 000	BEX	LDC6	
63	0074		0 101 011 000		LDC5	
64	0075		0 111 011 000	BX	LDC7	
65	0076		1 000 011 000	L8R	LDC8	
66	0077	0132	0 101 101 011		BRN	RETURN
67	0100		1 100 011 000	LOG	LDC12	
68	0101		0 110 011 000		LDC6	
69	0102		1 111 011 000		LDC15	
70	0103		0 110 011 000		LDC6	
71	0104		0 111 011 000		LDC7	
72	0105		0 111 011 000		LDC7	
73	0106		1 000 011 000		LDC8	
74	0107	0112	0 100 101 011		BRN	DONE
75	0110	0137	0 110 000 001		JSB	ACC
76	0111	0112	0 100 101 011		BRN	DONE
77	0112	0242	1 010 001 101	DONE	JSB	SHIFT
78	0113	0123	0 101 001 111	DONE1	BRN	WAIT
79	0114		1 001 101 100	DONE3	YP9	
80	0115	0127	0 101 011 111		BRN	DONE2
81	0116		1 010 010 000	LIST	ROM 5	
82	0117		0 000 111 100	DONE9	PLS	
83	0120		0 000 111 100		PLS	
84	0121	0243	1 010 010 001		JSB	SHIFT1
85	0122	0113	0 100 101 111		BRN	DONE1
86	0123		1 010 010 000	WAIT	ROM 5	
87	0124		0 110 011 000	BC	LDC6	
88	0125	0233	1 001 101 111		BRN	L3R
89	0126	0114	0 100 110 011		BRN	DONE3
90	0127		1 010 101 100	DONE2	YP10	
91	0130	0117	0 100 111 111		BRN	DONE9
92	0131	0116	0 100 111 011		BRN	LIST
93	0132		1 100 010 000	RETURN	ROM 6	
94	0133		0 110 011 000	CHS	LDC6	
95	0134		1 000 011 000		LDC8	
96	0135		0 111 011 000		LDC7	
97	0136		0 011 011 000		LDC3	
98	0137	0112	0 100 101 011		BRN	DONE
99	0140		0 110 011 000	ACC	LDC6	
100	0141		0 001 011 000		LDC1	
101	0142		0 110 011 000		LDC6	
102	0143		0 011 011 000		LDC3	
103	0144		0 110 011 000	C	LDC6	
104	0145		0 011 011 000		LDC3	
105	0146		0 000 110 000		RETURN	
106	0147		0 110 010 000	LCIO2	ROM 3	CONTINUE ROUTINE
107	0150	0137	0 110 000 001	BACC/	JSB	ACC
108	0151	0011	0 000 100 111		BRN	/
109	0152	0137	0 110 000 001	BACC*	JSB	ACC
110	0153	0012	0 000 101 011		BRN	AST
111	0154		0 000 000 000		DUMMY	
112	0155	0070	0 011 100 011	S/	BRN	B/

TYPEWRITER INTERFACE LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern			
113	0156	0022	0 001 001 011	SSTO	BRN	BSTO
114	0157	0063	0 011 001 111	SD	BRN	BD
115	0160	0056	0 010 111 011	SLE	BRN	BLE
116	0161	0030	0 001 100 011	SRND	BRN	BRND
117	0162	0040	0 010 000 011	SCL	BRN	BCL
118	0163	0050	0 010 100 011	SPRT	BRN	BPRT
119	0164	0075	0 011 110 111	SX	BRN	BX
120	0165	0124	0 101 010 011	SC	BRN	BC
121	0166	0305	1 100 010 111	SJMP	BRN	BJMP
122	0167	0203	1 000 001 111	SLBL	BRN	BLBL
123	0170	0213	1 000 101 111	SSUB	BRN	BSUB
124	0171	0073	0 011 101 111	SEX	BRN	BEX
125	0172	0223	1 001 001 111	SIF	BRN	BIF
126	0173	0230	1 001 100 011	SDS	BRN	BDS
127	0174		0 110 011 000	SFLG	LDC6	
128	0175		0 110 011 000		LDC6	
129	0176		0 110 011 000		LDC6	
130	0177		1 100 011 000		LDC12	
131	0200		0 110 011 000		LDC6	
132	0201		0 111 011 000		LDC7	
133	0202	0132	0 101 101 011		BRN	RETURN
134	0203		0 101 000 100	BLBL	SS5	
135	0204		0 110 011 000		LDC6	
136	0205		1 100 011 000		LDC12	
137	0206		0 110 011 000		LDC6	
138	0207		0 010 011 000		LDC2	
139	0210		0 110 011 000		LDC6	
140	0211		1 100 011 000		LDC12	
141	0212	0132	0 101 101 011		BRN	RETURN
142	0213		0 111 011 000	BSUB	LDC7	
143	0214		0 011 011 000		LDC3	
144	0215		0 111 011 000		LDC7	
145	0216		0 101 011 000		LDC5	
146	0217		0 110 011 000		LDC6	
147	0220		0 010 011 000		LDC2	
148	0221		0 110 000 100	BLBL1	SS6	
149	0222	0132	0 101 101 011		BRN	RETURN
150	0223		0 110 011 000	BIF	LDC6	
151	0224		1 001 011 000		LDC9	
152	0225		0 110 011 000		LDC6	
153	0226		0 110 011 000		LDC6	
154	0227	0132	0 101 101 011		BRN	RETURN
155	0230		0 110 011 000	BDS	LDC6	
156	0231		0 100 011 000		LDC4	
157	0232		0 111 011 000		LDC7	
158	0233		0 011 011 000	L3R	LDC3	
159	0234	0132	0 101 101 011		BRN	RETURN
160	0235		1 010 010 000	CR/LF2	ROM 5	
161	0236	0337	1 101 111 111		BRN	TABC
162	0237		0 000 000 000		DUMMY	
163	0240		0 000 000 000		DUMMY	
164	0241		0 000 000 000		DUMMY	
165	0242		0 000 000 000		DUMMY	
166	0243		0 111 110 000	SHIFT	CTT	
167	0244		0 101 000 000	SHIFT1	IS2	
168	0245		0 100 001 000		SLT	
169	0246		0 100 001 000		SLT	
170	0247		0 100 001 000		SLT	
171	0250		0 100 001 000		SLT	
172	0251		0 100 001 000		SLT	

TYPEWRITER INTERFACE LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
173	0252		0 100 001 000		SLT		
174	0253		0 100 001 000		SLT		
175	0254		0 100 001 000		SLT		
176	0255		0 100 101 000		IXT		
177	0256		1 001 000 000		IS1		
178	0257		0 000 110 000		RETURN		
179	0260		0 110 001 110		W,CTA		
180	0261		0 001 001 100		PT1		
181	0262		0 001 011 000		LDC1		
182	0263		0 110 011 000		LDC6		
183	0264		1 110 101 110		W,AXC		
184	0265		1 101 001 110		W,AMCA		
185	0266		1 101 111 010		XS,AM1A		
186	0267		0 100 001 110		W;SLA		
187	0270		0 100 001 110		W,SLA		
188	0271		1 011 001 010		X,SRA		
189	0272		1 101 111 010		XS,AM1A		
190	0273		0 000 111 100	FBSHIFT	PLS		
191	0274		0 100 001 110		W,SLA		
192	0275		0 100 101 100		YP4		
193	0276	0273	1 011 101 111		BRN	FBSHIFT	
194	0277		1 011 001 100		PT11		
195	0300		1 110 101 110		W,AXC		
196	0301		0 110 011 000		LDC6		
197	0302		0 110 011 000		LDC6		
198	0303		0 101 001 100		PT5		
199	0304	0112	0 100 101 011		BRN	DONE	
200	0305		0 110 011 000	BJMP	LDC6		
201	0306		1 010 011 000		LDC10		
202	0307		0 110 011 000		LDC6		
203	0310		1 101 011 000		LDC13		
204	0311		0 111 011 000		LDC7		
205	0312		0 000 011 000		LDC0		
206	0313	0221	1 001 000 111		BRN	BLBL1	
207	0314		0 000 000 000		DUMMY		
208	0315		0 000 000 000		DUMMY		
209	0316		0 010 101 000	FLG2	CXM		RESTORE FLAG AND DATA
210	0317		0 100 010 000		ROM 2		GO TO ROM 2 TO CONTINUE TESTS
211	0320		0 010 101 000		CXM		EXCHANGE FLAG AND DATA
212	0321		0 001 111 110		S,CM1		DOES OP LEVEL = 0?
213	0322	0316	1 100 111 011		BRN	FLG2	NO, CONTINUE TESTS
214	0323		0 111 001 100		PT7		YES, SET POINTER TO LOAD FLAG
215	0324		1 001 011 000	SETFLG	LDC9		SET FLAG
216	0325		0 010 101 000		CXM		RESTORE FLAG AND DATA
217	0326	0067	0 011 011 111		BRN	SUPVR	RETURN TO SUPERVISOR
218	0327	0317	0 110 000 001	BACC–	JSB	ACC	
219	0330	0047	0 010 011 111		BRN	–	
220	0331	0137	0 110 000 001	BACC+	JSB	ACC	
221	0332	0037	0 001 111 111		BRN	+	
222	0333	0331	1 101 100 111	ACC+	BRN	BACC+	
223	0334	0327	1 101 011 111	ACC–	BRN	BACC–	
224	0335	0152	0 110 101 011	ACC*	BRN	BACC*	
225	0336	0150	0 110 100 011	ACC/	BRN	BACC/	
226	0337		0 111 001 100	TABC	PT7		SET POINTER TO TAB FLAG
227	0340		0 001 100 010		P,CM1		IS FLAG SET?
228	0341	0363	1 111 001 111		BRN	TAB	YES, OUTPUT A TAB
229	0342	0354	1 110 110 011		BRN	PRETURN	NO, RETURN TO ROUTINE
230	0343		0 111 011 000	RET	LDC7		
231	0344		0 010 011 000		LDC2		
232	0345		0 110 011 000		LDC6		

TYPEWRITER INTERFACE LISTING (ROM 7) Continued

Line #	Curr Addr	Bran Addr	Operation Code Bit Pattern				
233	0346		0 101 011 000		LDC5		
234	0347		0 111 011 000		LDC7		
235	0350		0 100 011 000		LDC4		
236	0351	0112	0 100 101 011		BRN	DONE	
237	0352		0 000 000 000		DUMMY		
238	0353		0 000 000 000		DUMMY		
239	0354		0 110 010 000	PRETURN	ROM 3		DO NOT TAB
240	0355		0 110 100 100		RS6		RESET SCIENTIFIC NOTATION FLAG
241	0356		1 010 101 000		MTC		RECALL TFLAG FOR FUNCTION TEST
242	0357		1 000 001 100	CR/LF	PT8		SET POINTER TO CR/LF FLAG
243	0360		0 001 100 010		P,CM1		IS FLAG SET?
244	0361	0370	1 111 100 011		BRN	CR/LF1	YES, OUTPUT CR/LF
245	0362	0337	1 101 111 111		BRN	TABC	NO, CHECK IF TAB FLAG IS SET
246	0363		0 101 000 000	TAB	IS2		PREPARE TO LOAD CODE FOR TAB
247	0364	0037	0 010 000 001		LIO 100		LOAD TAB CODE
248	0365		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
249	0366		1 001 000 000		IS1		PREPARE TO RETURN
250	0367	0147	0 110 011 111		BRN	LCIO2	OUTPUT TAB
251	0370		0 101 000 000	CR/LF1	IS2		PREPARE TO LOAD CODE FOR CR/LF
252	0371	0002	0 000 001 101		LIO	006	LOAD CODE FOR CR/LF
253	0372		0 101 101 000		SRI		SHIFT CODE TO OUTPUT POSITION
254	0373		1 001 000 000		IS1		PREPARE TO RETURN
255	0374	0235	1 001 110 111		BRN	CR/LF2	OUTPUT CR/LF
256	0375		0 110 011 000	LOGX	LDC6		
257	0376	0100	0 100 000 011		BRN	LOG	
258	0377	0143	0 110 010 001	CALL	JSB	C	

CALCULATOR OPERATION

The basic calculator contains three register locations into which numbers are either entered for immediate use or are stored for later use. These are the X-register in the arithmetic and register circuit 32 and the C-register and RT-register in the data storage assembly 36.

Each number which the user enters from the keyboard is placed in the X-register. If the calculator is configured with the plug-in LED display unit, the number is displayed as it is entered. The X-register also receives the result of each mathematical operation.

Each time the EQUALS key is actuated, a result is generated, which is placed in the X-register and also added to the contents of the RT-register for accumulating the result total.

The C-register may be used for storing any constant for later use in making calculations. Whenever the STORE key is actuated, the contents of the X-register are duplicated (stored) in the C-register. Any constant stored in the C-register remains unaltered until either another constant is stored therein or the calculator is switched off. The RECALL key is used to bring the contents of the C-register back into the X-register.

Actuating the CLEAR key erases the contents of the X and RT registers and causes the word CLEAR to be printed on the output printer. CLEAR operations leave the contents of the C-register unaltered.

Actuating the CANCEL ENTRY key erases the X-register but leaves the C and RT registers unaltered.

When operating power to the calculator is removed all storage registers are erased.

In addition to printing CLEAR and tallying keyboard operations, the calculator informs the user of his operating errors by printing diagnostic notes when an error has been made. A table of diagnostic notes appears later herein. Some errors may be readily corrected, while others may require that the CLEAR key be actuated and the problem begun again.

The four basic arithmetic operations comprising addition, subtraction, multiplication, and division may be easily accomplished with the calculator. The first operand is entered; the desired operation key is actuated; the second operand is entered; and then the EQUALS key is actuated to execute the calculation. The result is placed in the X-register. When performing calculations involving more than one operator, each successive operation uses the result of the previous operation.

A red busy light on the keyboard flashes whenever the calculator is performing a function and cannot accept key actuations. When the calculator is executing a library program stored in a plug-in ROM or PROM, the busy light may remain on for several seconds. This light simply serves to remind the operator that the calculator will ignore any key actuations which occur during the time it is busy.

Negative numbers are entered by first keying in the number and then changing its sign. The sign is changed by successively actuating the SHIFT key and the LEFT PARENTHESIS key. Negative numbers are printed in red on the output printer.

As discussed above, the calculator performs successive mathematical operations by using the result of the previous operation. By using the LEFT PARENTHESIS and RIGHT PARENTHESIS keys, operations may be grouped together just as is done in standard mathematical notation involving parentheses. Use of parentheses in expressions eliminates the need for storing and recalling intermediate results. The output printer prints the result of each operation within parentheses at the time each right parenthesis is entered. Parentheses may be nested up to and including five levels.

The EXPONENT key (↑) is used to raise any number to any power. The number to be raised is entered, followed by the EXPONENT key, followed by the power.

The PERCENT key may be used in conjunction with all four arithmetic operators to calculate, for example, a percentage of a number in the X-register or to further use a percentage of a number in the X-register within a calculation. For example, if it is desired to calculate 6% of 39.95, the user would enter 39.95, followed by the MULTIPLY key, followed by 6, followed by the PERCENT key, terminated by the EQUALS key, to give a result of 2.40. As a further example, if the user wishes to discount or subtract 20% from 80, he would first enter 80, followed by the SUBTRACT key, followed by 20, followed by the PERCENT key,

terminated by the EQUALS key, to give a result of 64. As a final example, if it is desired to calculate the simple interest for a term of 2 years on $500 at 7%, the user would first enter 500, followed by the MULTIPLY key, followed by 7, followed by the PERCENT key, followed by the MULTIPLY key, followed by 2, terminated by the EQUALS key, to give a result of $70.

The output printer unit is associated with several keys which control its operation. The PRINT OFF (PRT OFF) key suppresses tallying of keyboard operations on the printer, but allows printing of the diagnostic notes plus CLEAR. Actuation of the PRINT key prints the current contents of the X-register, regardless of whether the PRINT OFF mode is in effect. The symbol ≠ accompanies each number printed by actuating the PRINT key. The PAPER key is used to quickly advance the printer paper. When the calculator is turned on the display and print format is automatically set to ROUND 2. This means that each number displayed and/or printed is shown with two digits to the right of the decimal point. This format can be altered to indicate from zero to six decimal point digits by actuating the ROUND [RND ()] key followed by the appropriate number. Printed and displayed numbers may be shown in floating point, or scientific notation by actuating the ROUND key followed by the DECIMAL POINT key.

Some of the calculator keys represent dual functions, distinguished by whether actuation has been preceded by actuation of the SHIFT key. Actuation of the SHIFT key followed by the EXPONENT key operates to calculate the reciprocal of the contents of the X-register. Actuation of the SHIFT key followed by the DIVIDE key results in calculating the common logarithm of the number in the X-register. Actuation of the SHIFT key followed by the MULTIPLY key is used to calculate the natural logarithm of the contents of the X-register.

Actuation of the SHIFT key followed by the ADD key operates to raise the Naperian logarithm base *e* to the power indicated by the contents of the X-register. Actuation of the SHIFT key followed by the SUBTRACT key acts to divide the number in the X-register by 12.

When solving problems involving the entry of many numbers in the same form, such as dollars and cents, the AUTO DECIMAL POINT key may be used to automatically place the decimal point at a specified position in each number entry. This eliminates physically depressing the DECIMAL POINT key at the proper point during entry of each number. The place at which the decimal point is automatically located is specified by the current rounding format. For instance, if the calculator is operating in ROUND 2 format and the automatic decimal point feature is being used, the decimal point will be placed so that two digits will be indicated to the right thereof.

The RT-register accumulates results that are placed in the X-register each time the EQUALS key is actuated. To recall the contents of the RT-register to the X-register, and at the same time display and print the new contents of the X-register, the user simply actuates the SHIFT key followed by the EQUALS key.

As described in detail earlier in this specification, the user may plug into the calculator optional data storage units. This option is available to expand the single C-register of the basic calculator in increments of 30 or 100 registers. The extra registers are numbered zero through twenty-nine or zero through ninety-nine. To store the data currently in the X-register into one of the registers of the data storage option the SHIFT key is first actuated, followed by the STORE key, followed by the desired register number, terminated by the EQUALS key. To recall a number from optional data storage into the X-register, the same key sequence is followed except that the RECALL key is actuated instead of the STORE key. It is possible to obtain a printed listing of all the data stored in the optional registers by means of the following key sequence: SHIFT, START PROGRAM (START PRGM), PRINT, EQUALS. The register contents are printed in blocks of ten, beginning at register zero. To erase all of the optional data storage registers, the following key sequence is observed: SHIFT, START PROGRAM, ZERO, EQUALS. The optional data storage registers may be incremented or decremented, respectively, by actuating the SHIFT key, followed by the START PROGRAM key, followed by either the ADD key or the SUBTRACT key.

As discussed above, a plug-in read-only memory (ROM) or programmable read-only memory (PROM) may be employed with the calculator. Such a ROM or PROM may contain one or more programs written in a user level language which may be run from the calculator keyboard. The programs stored within a plug-in

ROM or PROM may be generated by either the user or factory personnel by means of instruction routines stored within ROM circuit A—Ø of the basic calculator ROM group 34. Each of these routines has a starting address in ROM circuit A—Ø associated with it. When generating a program for implementation in a plug-in ROM or PROM, the starting address of each routine to be performed is placed in the ROM or PROM bit patterns in the order in which such routines are to be performed by the program. The available routines from which the user may select in creating a program to be implemented in a plug-in ROM or PROM include many which are not represented as keyboard functions executable in the manual mode. The table below lists each available instruction, together with its starting address in ROM circuit A—Ø of the basic calculator ROM group 34.

INTRUCTION ROUTINE STARTING ADDRESS	MNEMONIC	INSTRUCTION DESCRIPTION
064	START	START PROGRAM EXECUTION
012	/	FLOATING POINT DIVISION
013	*	FLOATING POINT MULTIPLICATION
014	STOX	POWER FUNCTION (X^Y)
016	)	RIGHT PARENTHESIS
020	=	EQUAL
022	.	DECIMAL POINT
023	Ø	DIGIT ZERO
026	(	LEFT PARENTHESIS
032	3	DIGIT 3
033	2	DIGIT 2
034	1	DIGIT 1
036	%	PERCENT
040	+	FLOATING POINT ADDITION
042	6	DIGIT 6
043	5	DIGIT 5
044	4	DIGIT 4
046	CRCL	RECALL DATA FROM USER STORAGE REGISTER TO X–REGISTER
050	–	FLOATING POINT SUBTRACTION
052	9	DIGIT 9
053	8	DIGIT 8
054	7	DIGIT 7
056	STO	STORE DATA INTO USER STORAGE REGISTER
062	RUN	CONTINUE PROGRAM EXECUTION
063	LE	SAT LAST ENTRY FLAG

INSTRUCTION ROUTINE STARTING ADDRESS	MNEMONIC	INSTRUCTION DESCRIPTION
066	CANCEL	CANCEL ENTRY
070	SHIFT	SHIFT
072	RND	ROUND
073	CLR	CLEAR WORKING REGISTERS
076	PRTX	PRINT DATA IN X–REGISTER
150	JMP–	JUMP RELATIVE BACKWARD
107	NOP	PROGRAMMED PAUSE
131	CHS	CHANGE SIGN OF DATA
207	LBL	LABEL HEAD BEGINNING
152	JMP+	JUMP RELATIVE FORWARD
224	SUBLBX	JUMP SUBROUTINE TO COMPUTED LABEL
225	SUBLBL	JUMP SUBROUTINE TO LABEL
226	SUB	JUMP SUBROUTINE TO ADDRESS
237	TBL	CALL FUNCTION BLOCK TABLE FUNCTION ()
240	FØ1	CALL FUNCTION BLOCK KEY FUNCTION #1
241	FØ2	CALL FUNCTION BLOCK KEY FUNCTION #2
242	FØ3	CALL FUNCTION BLOCK KEY FUNCTION #3
243	FØ4	CALL FUNCTION BLOCK KEY FUNCTION #4
224	FØ5	CALL FUNCTION BLOCK KEY FUNCTION #5
245	FØ6	CALL FUNCTION BLOCK KEY FUNCTION #6
246	FØ7	CALL FUNCTION BLOCK KEY FUNCTION #7
247	FØ8	CALL FUNCTION BLOCK KEY FUNCTION #8
250	FØ9	CALL FUNCTION BLOCK KEY FUNCTION #9
251	F1Ø	CALL FUNCTION BLOCK KEY FUNCTION #10
252	F11	CALL FUNCTION BLOCK KEY FUNCTION #11
253	F12	CALL FUNCTION BLOCK KEY FUNCTION #12
254	F13	CALL FUNCTION BLOCK KEY FUNCTION #13
255	F14	CALL FUNCTION BLOCK KEY FUNCTION #14
256	F15	CALL FUNCTION BLOCK KEY FUNCTION #15
263	STOP	STOP PROGRAM EXECUTION
267	1/X	RECIPROCAL OF X–REGISTER
270	EXP	EXPONENTIAL FUNCTION (e^X)

INSTRUCTION ROUTINE STARTING ADDRESS	MNEMONIC	INSTRUCTION DESCRIPTION
275	IFØ	IF DATA IS ZERO TEST
300	STEP	PROGRAM EXECUTION STEP
312	IF+	IF DATA IS POSITIVE TEST
316	IF-	IF DATA IS NEGATIVE
322	IFLE	IF LAST ENTRY FLAG SET TEST
140	PRTACC	PRINT GRAND TOTAL FROM ACCUMULATOR
325	PRTDS	LIST DATA STORAGE OPTION REGISTERS
326	CLDS	CLEAR OPTIONAL DATA STORAGE REGISTER
327	DS+	INCREMENT DATA STORAGE GIVEN BY (C)
330	DS-	DECREMENT DATA STORAGE GIVEN BY (C)
331	STO	STORE DIRECT IN DATA STORAGE ADDRESS ()
332	ACC+	CHANGE STORE TO ACCUMULATE +
333	ACC-	CHANGE STORE TO ACCUMULATE -
334	ACC*	CHANGE STORE TO ACCUMULATE X
335	ACC/	CHANGE STORE TO ACCUMULATE ÷
336	RCL	RECALL FROM DATA STORAGE ADDRESS ()
337	EXCH	EXCHANGE WITH DATA AT ADDRESS ()
342	RET	RETURN FROM SUBROUTINE
346	SFLG	SET FLAG ()
347	IFFLG	IF FLAG () SET TEST
353	JMPLBX	GO TO COMPUTED LABEL
354	JMPLBL	GO TO LABEL
355	JMP	GO TO ADDRESS
362	X/12	X-REGISTER DIVIDED BY 12
363	ADV	PRINTER PAPER ADVANCE
373	END	PROGRAM END
374	LOGX	COMMON LOGARITHM
375	LNX	NATURAL LOGARITHM
376	CALL	CALL I/O UNIT ()

Even though a plug-in ROM or PROM is employed, the calculator may still be used in the manual mode. A plug-in ROM or PROM may contain any number of separate programs, limited only by the capacity of the ROM or PROM. Separate programs may be designated by any number between zero and nine

hundred. To run a program which is the only one contained in a particular ROM or PROM it is only necessary to actuate the START PROGRAM key. If the program contains halts for data entry from the keyboard the user must, after each data entry, actuate the RUN/STOP key. If the ROM or PROM currently employed contains several programs, the user may begin execution of any one of them by actuating the START PROGRAM key, followed by the program designation number, followed by actuation of the RUN/STOP key. During execution of a ROM/PROM program successive actuations of the RUN/STOP key will first halt and then resume program execution.

In implementing a program in plug-in ROM or PROM, those instruction routines residing in ROM circuit A—Ø of the basic calculator may be utilized, as shown and described above. In addition, when one of the plug-in keyboard function blocks is employed with the calculator, additional instruction routines residing within each function block are also available for use in constructing a program in plug-in ROM or PROM.

The instruction routines available in the MATH/USER DEFINABLE plug-in keyboard function block are shown in the table below. These functions are called by the function block table call routine TBL, located at address 237 of ROM circuit A—Ø and shown in the table above, by specifying the key code accompanying each instruction routine listed below.

INSTRUCTION ROUTINES IN THE TABLE OF MATH/USER DEFINABLE FUNCTION BLOCK

KEY CODE	INSTRUCTION ROUTINE
00	ANGLE ENTRY (DEGREES, MINUTES, SECONDS)
01	SINE
02	COSINE
03	TANGENT
04	ARC SINE
05	ARC COSINE
06	ARC TANGENT
07	SQUARE ROOT OF X–REGISTER
09	X^2
10	π (PI)
11	INTEGER OF X
12	ABSOLUTE VALUE OF X
13	RECTANGULAR TO POLAR CONVERSION
14	POLAR TO RECTANGULAR CONVERSION
15	DISPLAY DATA STORAGE REGISTER Ø
16	VECTOR ACCUMULATE +
17	VECTOR ACCUMULATE -
18	RECALL VECTOR ACCUMULATE ±
19	CLEAR VECTOR ACCUMULATE ±

D-28

INSTRUCTION ROUTINES IN THE TABLE OF MATH/USER DEFINABLE FUNCTION BLOCK

KEY CODE	INSTRUCTION ROUTINE
20	CONVERT DEGREES, MINUTES, SECONDS TO DECIMAL DEGREES
21	CONVERT DECIMAL DEGREES TO DEGREES, MINUTES, SECONDS
22	CONVERT GRADS TO DECIMAL DEGREES
23	CONVERT DECIMAL DEGREES TO GRADS
24	CONVERT RADIANS TO DECIMAL DEGREES
25	CONVERT DECIMAL DEGREES TO RADIANS
26	CONVERT MILS TO DECIMAL DEGREES
27	CONVERT DECIMAL DEGREES TO MILS
28	ENTER EXPONENT
30	PRINT DATA STORAGE REGISTERS Ø & 1 WITH X, Y LABELS
31	PRINT DATA STORAGE REGISTERS Ø & 1 WITH R, A LABELS
4N	STORE IN N* $0 \leq N \leq 9$
5N	RECALL FROM N* $0 \leq N \leq 9$

* Where N is one of the 10 data storage registers within the function block.

In addition to their availability in constructing programs for plug-in ROM or PROM, the above listed instruction routines may be called from the basic calculator keyboard in the manual mode. This is accomplished by successively actuating the SHIFT key and the DECIMAL POINT key followed by the key code of the desired instruction routine (function).

Each of the fifteen keys on the MATH/USER DEFINABLE plug-in function block is associated with a particular address in a plug-in ROM or PROM. Therefore, a program written in a ROM or PROM may be arranged so that its starting address corresponds to that associated with one of the function block keys. Such a program may then be executed by simply depressing the related function block key. These keys have removable transparent caps, into which the user may insert labels identifying the program or function associated therewith.

DIAGNOSTIC NOTES

As discussed above, user errors occurring during operation of the calculator are indicated by means of a printed note. A listing of these notes with corresponding error description is given below.

NOTE Ø0 TOO MANY SUBROUTINE CALLS OR RETURNS

NOTE Ø4 FIVE PARENTHESES LEVELS ALREADY ASSIGNED

NOTE Ø5	RIGHT PARENTHESIS NOTE PRECEDED BY LEFT PARENTHESIS
NOTE Ø8	NEGATIVE NUMBER RAISED TO A NON-INTEGER POWER
NOTE 9	LOG OR LN OF A NEGATIVE NUMBER
NOTE 11	BEFORE RESULT MEANS REGULAR OVERFLOW AFTER RESULT MEANS RT-REGISTER OVERFLOW
NOTE 12	LOG OR LN OF ZERO
NOTE 15	ZERO TO A NEGATIVE POWER
NOTE 16	ATTEMPTED DIVISION BY ZERO
NOTE 20	LABEL NOT FOUND
NOTE 21	IØ UNIT NOT FOUND PROGRAM MEMORY EXCEEDED PROGRAM MEMORY NOT READY DATA STORAGE SOFTWARE NOT READY FUNCTION BLOCK NOT READY
NOTE 22	RIGHT PARENTHESIS AFTER AN OPERATION
NOTE 24	EQUAL SIGN INSIDE PARENTHESES
NOTE 26	DATA STORAGE ADDRESS NOT FOUND

It will be appreciated that the described programmed calculator has more capability and flexibility than conventional calculators which are dedicated for solving a narrow range of problems and is smaller and less expensive than conventional programmable calculators.

It will also be appreciated that in the described calculator, programs stored in a read-only memory or programmable read-only memory are written at the user language level rather than the microprogram level, thereby allowing the user to generate or alter such programs without knowledge of the microprogramming language of the calculator.

It will also be appreciated that in the described calculator, the user can enlarge the keyboard thereof by plugging into said keyboard a function block containing a plurality of keys together with associated read-only memory, said keys either representing predefined functions or functions definable by the user.

It will also be appreciated that in the described calculator, a definable plug-in function block for enlarging the keyboard of the calculator contains a table of microprogram functions and such functions, together with additional functions from a microprogram library contained elsewhere in the calculator, may be selectively employed to construct functions or programs to be associated with particular ones of a plurality of keys contained within said function block.

It will also be appreciated that in the described calculator, a program stored in a plug-in read-only memory or a plug-in programmable read-only memory can call not only microprogram functions stored within the calculator mainframe, but also microprogram functions stored within a plug-in keyboard function block employed in the calculator.

It will also be appreciated that in the described calculator, each key of a user-definable keyboard function block represents a function or program defined, by a read-only memory or programmable read-only memory currently plugged into the calculator, as a sequence of microprogram subroutines which are contained within the calculator mainframe and/or the user-definable function block and which may or may not be represented as keyboard functions.

It will also be appreciated that in the described calculator a user-level program written in a read-only memory or a programmable read-only memory may be run without the availability of keys associated with program writing.

It will also be appreciated that in the described calculator each peripheral input/output unit is interfaced to the calculator by means of a single printed circuit board which contains all necessary hardware and software for driving the peripheral.

It will also be appreciated that with the described calculator the user may designate an automatic decimal point mode for automatically placing the decimal point in entered data at a preselected position.

The user can also employ a PER CENT key in combination with data and one or more of the four arithmetic operators.

In addition the user may enter various items of data, each followed by an arithmetic operator and may enter an equal sign following any of the entered arithmetic operators for calculating the result to that point.

WHAT WE CLAIM IS:—

1. An electronic calculator comprising:

a keyboard input unit for providing the user with a plurality of keyboard functions and for entering into the calculator information used to initiate selected ones of those keyboard functions and data used in making selected calculations to perform the selected keyboard functions;

a memory unit for storing routines and subroutines of instructions to be performed by the calculator in making the selected calculations to perform the selected keyboard functions;

a modular keyboard section, including a read-only memory containing additional routines and subroutines of instructions, for providing the user with additional keyboard functions and for entering into the calculator information used to initiate selected ones of those additional keyboard functions;

processing means responsive to information from the keyboard input unit and to operating states within the calculator itself for selectively performing one or more of the routines and subroutines stored in the memory unit to make the selected calculations employing data entered from the keyboard input unit;

said processing means being responsive to information from the modular keyboard section and to operating states within the calculator itself for selectively performing one or more of the additional routines and subroutines of instructions contained in the read-only memory to make selected calculations employing data entered from at least one of the keyboard input unit and the modular keyboard section; and

output means for providing an output indication of the results of the calculations performed by the processing means.

2. An electronic calculator as in claim 1 wherein:

said keyboard input means includes a receptacle into which said modular keyboard section may be removably plugged; and

said keyboard input means and said modular keyboard section include electrical connection means for electrically connecting said keyboard input means and said modular keyboard section.

3. An electronic calculator as in either one of claims 1 and 2 wherein said modular keyboard section includes a plurality of keys each of which represents a predetermined function.

4. An electronic calculator as in either one of claims 1 and 2 wherein said modular keyboard section includes a plurality of keys each of which may be associated with a function defined by the user.

5. An electronic calculator according to any one of the claims 1 to 4 and further comprising:

interface means for coupling a selected external input/output peripheral unit to the calculator, said interface means including logic means for transferring data between the calculator and the selected input/output peripheral unit and further including microprogrammed read-only memory means for enabling said processing means to perform logic operations associated with the selected input/output peripheral unit.

6. An electronic calculator according to claim 1 constructed, arranged and adapted to operate substantially as hereinbefore described with reference to the accompanying drawings.

It will also be appreciated that in the described calculator each peripheral input/output unit is interfaced to the calculator by means of a single printed circuit board which contains all necessary hardware and software for driving the peripheral.

It will also be appreciated that with the described calculator the user may designate an automatic decimal point mode for automatically placing the decimal point in entered data at a preselected position.

The user can also employ a PER CENT key in combination with data and one or more of the four arithmetic operators.

In addition the user may enter various items of data, each followed by an arithmetic operator and may enter an equal sign following any of the entered arithmetic operators for calculating the result to that point.

WHAT WE CLAIM IS:—

1. An electronic calculator comprising:

a keyboard input unit for providing the user with a plurality of keyboard functions and for entering into the calculator information used to initiate selected ones of those keyboard functions and data used in making selected calculations to perform the selected keyboard functions;

a memory unit for storing routines and subroutines of instructions to be performed by the calculator in making the selected calculations to perform the selected keyboard functions;

a modular keyboard section, including a read-only memory containing additional routines and subroutines of instructions, for providing the user with additional keyboard functions and for entering into the calculator information used to initiate selected ones of those additional keyboard functions;

processing means responsive to information from the keyboard input unit and to operating states within the calculator itself for selectively performing one or more of the routines and subroutines stored in the memory unit to make the selected calculations employing data entered from the keyboard input unit;

said processing means being responsive to information from the modular keyboard section and to operating states within the calculator itself for selectively performing one or more of the additional routines and subroutines of instructions contained in the read-only memory to make selected calculations employing data entered from at least one of the keyboard input unit and the modular keyboard section; and

output means for providing an output indication of the results of the calculations performed by the processing means.

2. An electronic calculator as in claim 1 wherein:

said keyboard input means includes a receptacle into which said modular keyboard section may be removably plugged; and

said keyboard input means and said modular keyboard section include electrical connection means for electrically connecting said keyboard input means and said modular keyboard section.

3. An electronic calculator as in either one of claims 1 and 2 wherein said modular keyboard section includes a plurality of keys each of which represents a predetermined function.

4. An electronic calculator as in either one of claims 1 and 2 wherein said modular keyboard section includes a plurality of keys each of which may be associated with a function defined by the user.

5. An electronic calculator according to any one of the claims 1 to 4 and further comprising:

interface means for coupling a selected external input/output peripheral unit to the calculator, said interface means including logic means for transferring data between the calculator and the selected input/output peripheral unit and further including microprogrammed read-only memory means for enabling said processing means to perform logic operations associated with the selected input/output peripheral unit.

6. An electronic calculator according to claim 1 constructed, arranged and adapted to operate substantially as hereinbefore described with reference to the accompanying drawings.

* C L M S *

ERIC POTTER & CLARKSON,
Chartered Patent Agents,
Kingsway House,
Kingsway,
London, WC2B 6QX

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1976.
Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

