NASA CR-165,875

NASA Contractor Report 165875

NASA-CR-165875
19840021429

Programming For Energy Monitoring/Display System in Multicolor Lidar System Research

LIBRARY COPY

APR 27 1982

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

Ramon C. Alvarado, Jr., Robert J. Allen, and Gary E. Copeland

OLD DOMINION UNIVERSITY RESEARCH FOUNDATION
Norfolk, Virginia 23508-0369

Contract NCC1-32
March 1982

~~FOR EARLY DOMESTIC DISSEMINATION~~

~~Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.~~

~~Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.~~

~~Review for general release~~ March 31, 1984

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

NF01948

DEPARTMENT OF PHYSICS
SCHOOL OF SCIENCES AND HEALTH PROFESSIONS
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

Technical Report PTR-81-11

PROGRAMMING FOR ENERGY MONITORING/DISPLAY SYSTEM IN MULTICOLOR LIDAR SYSTEM RESEARCH

By

Ramon C. Alvarado, Jr.
Robert J. Allen

and

Gary E. Copeland, Principal Investigator

Progress Report
For the period June 15, 1980 - June 14, 1981

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Cooperative Agreement NCC1-32
E.V. Browell, Technical Monitor
Atmospheric Environmental Sciences Division

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508-0369

December 1981

N82-74122 #

This Page Intentionally Left Blank

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES - CONCLUDED

LIST OF FIGURES

LIST OF FIGURES - CONCLUDED

PROGRAMMING FOR ENERGY MONITORING/DISPLAY SYSTEM IN MULTICOLOR LIDAR SYSTEM RESEARCH

By

Ramon C. Alvarado, Jr.[1], Robert J. Allen[2], and Gary E. Copeland[3]

ABSTRACT

This report describes the Z80 microprocessor-based computer program that directs and controls the operation of the six-channel energy monitoring/display system that is a part of the NASA Multipurpose Airborne Differential Absorption Lidar (DIAL) System. The program is written in the Z80 assembly language[4] and is located on EPROM memories. All source and assembled listings of the main program, five subroutines, and two service routines along with flow charts and memory maps are included. A combinational block diagram shows the interfacing (including port addresses) between the six power sensors, displays, front panel controls, the main general purpose minicomputer, and this dedicated microcomputer system.

INTRODUCTION

The Z80 microprocessor-based program presented here has been written to direct and control the operation of the six-channel energy monitoring/display system that is a part of the NASA Multipurpose Airborne Differential Absorption Lidar System (or NASA DIAL System for short). This system incorporates remote laser monitoring to obtain measurements of atmospheric pollutants (SO_2, O_3, particulates). At present, the NASA DIAL System includes two frequency tunable dye lasers that are pumped by doubled Nd-YAG lasers which radiate at 0.53 μm. The dye laser outputs are passed through optical

[1] Graduate Research Assistant, Department of Physics, Old Dominion University, Norfolk, VA 23508.

[2] Research Professor, Department of Physics, Old Dominion University, Norfolk, VA 23508.

[3] Associate Professor, Department of Physics, Old Dominion University, Norfolk, VA 23508.

[4] Z80 is a trademark of Zilog, Inc., with whom the publisher is not associated.

crystals that double the frequency of the transmitted light to UV wavelengths. A small fraction of the output from each of the pump lasers (designated P1 and P2), dye lasers (designated D1 and D2), and UV doublers (designated X1 and X2) is converted into electrical signals by fast photodiodes, integrated to convert power into energy, stored temporarily in sample-and-hold amplifiers, and digitized by six A-to-D converters (fig. F1). The digital outputs from the A-to-D converters are processed by the Z80-CPU and associated logic as illustrated by the combinational block diagrams included as Appendix F. Further information regarding the NASA DIAL system can be found in references 1 to 4.

The computer program described here is written in Z80 assembly language* and is executed by the Z80 microprocessor while the laser transmitter/receiver system is operating. When executed, the program handles the energy data provided by the six A-to-D channels (fig. F1): channel 0 - pump laser P1, channel 1 - dye laser D1, channel 2 - UV doubling crystal X1, channel 3 - pump laser P2, channel 4 - dye laser D2, channel 5 - UV doubling crystal X2.

PROGRAM DEVELOPMENT

Objectives

As presently configured**, this program has four main objectives: (1) to input the energies of the laser and doubling crystal transmissions, (2) to calculate the average energy for each of the six channels for a selected number of firings, (3) to transfer the energy averaged data to the PDP-11/34 minicomputer, and (4) to display the average transmitted energy of any three selected channels on the multichannel display. In creating a program to meet these objectives, the constraints imposed by the firing rate of the laser system, along with the conversion times of the A-to-D converters, must be considered. The maximum firing rate of the system is 10 Hz, and the time required by the A-to-D converters to convert the six-channel signals into

*A summary of the Z80-CPU instruction set is included as Appendix E. A thorough description of the Z80 instruction set and assembly language programming can be found in the literature (refs. 5-8).

**Future configurations planned include (1) logging the PMT gain function generator unit "step gain switch" positions (fig. F5) and (2) diagnosing laser optical misalignments or failures.

their binary equivalents is 40 ms. There is a total of 100 ms following conversion in which to complete the objectives listed above. The following subsection briefly describes how the program presented in this report accomplishes the above requirements.

Basic Program Function

Basically, the Z80 programming for the energy monitoring/display system functions as follows: On startup, the energy-averaging switches are read. This data is used to determine N, which is used to calculate 2^N, which is the number of firings to be averaged. Next, for each firing of the laser system, the energies of the six channels are read and used to calculate the current or running average for each channel's energy per transmission based on the number of firings made so far during the current averaging run. When the desired number of transmissions to be averaged has been reached, the low byte of the average energy per transmission for channel 0 is sent to the PDP-11, whereupon a flag is asserted. If the PDP-11 acknowledges the reception of this data within 12 μs* after its transmission, the remainder of the energy average data is transferred. If the PDP-11 does not come back within the allotted time, there are no transfers of energy average data and the selection switches of the three-channel selectable display are read. The digital displays are then programmed to display the average energy per transmission for the selected three channels.

At this point in the program, for a normal operating sequence, the energy-averaging switches would again be read and a new averaging run would commence. The normal mode of operation, however, can be altered at any time by an $\overline{\text{NMI}}$ interrupt generated by the PDP-11. The details of such a procedure and of the regular workings of the program will be described later.

Memory Maps

The energy monitoring/display system program uses 2K of programmable, read only memory (PROM) and 2K of random access memory (RAM) as indicated in figure F6. The PROMs are assigned addresses 0 to 2047_{10} (00H - 7FFH), and the RAMs addresses 2048_{10} to 4095_{10} (800H - FFFH). How the available memory is utilized is shown by the memory maps (tables 1 and 2).

*This value may be altered after further evaluation.

Table 1. Memory map for two 1024 × 8 EPROMS.

ADDRESSES: 0-2047 (00H-07FFH)

ADDRESS DEC	ADDRESS HEX	# BYTES	DESCRIPTION
0	0		JUMP INSTRUCTION TO BEGINNING OF
2	2	3	MAIN PROGRAM
3	3		NOT USED
55	37	53	
56	38		JUMP TO INT. INTERRUPT SERVICE
58	3A	3	ROUTINE DATRNSFR
59	3B		NOT USED
101	65	43	
102	66		JUMP TO NMI INTERRUPT SERVICE
104	68	3	ROUTINE PDPSER
105	69		NOT USED
255	FF	151	
256	100		SUBROUTINE DIVIDE
280	118	25	
281	119		NOT USED
287	11F	7	
288	120		SUBROUTINE ADDREG
300	12C	13	
301	12D		NOT USED
319	13F	19	
320	140		SUBROUTINE REARRG
352	160	33	
353	161		NOT USED
367	16F	15	
368	170		SUBROUTINE SWSEL1
600	258	233	
601	259		NOT USED
639	27F	39	
640	280		SUBROUTINE SWSEL2
706	2C2	67	
707	2C3		NOT USED
735	2DF	29	
736	2E0		SERVICE ROUTINE DATRNSFR
782	30E	47	
783	30F		NOT USED
808	328	26	
809	329		SERVICE ROUTINE PDPSER
840	348	32	
841	349		THOUSN TABLE
842	34A	2	
843	34B		HUNDRD TABLE
844	34C	2	
845	34D		TENS TABLE
846	34E	2	
847	34F		ONES TABLE
848	350	2	
849	351		NOT USED
1023	3FF	175	
1024	400		PROGRAM MAIN
1651	673	628	
1652	674		NOT USED
2047	7FF	396	

A.

Table 2. Memory map for four 1024 × 4 RAMS.

MEMORY MAP FOR FOUR 1024 X 4 RAMS
****

ADDRESSES: 2048-4095 (800H-FFFH)

ADDRESS DEC	ADDRESS HEX	# BYTES	DESCRIPTION
2048	800		P1ESUM
2050	802	3	P1ESM3
2051	803		D1ESUM
2053	805	3	D1ESM3
2054	806		X1ESUM
2056	808	3	X1ESM3
2057	809		P2ESUM
2059	80B	3	P2ESM3
2060	80C		D2ESUM
2062	80E	3	D2ESM3
2063	80F		X2ESUM
2065	811	3	X2ESM3
2066	812		NOT USED
2068	814	3	
2069	815	1	NADDR
2070	816	1	NOT USED
2071	817		NUMFIR
2072	818	2	
2073	819		NOT USED
2079	81F	7	
2080	820		DECMAL
2083	823	4	DECML4
2084	824		NOT USED
2086	826	3	
2087	827		BINVAL
2088	828	2	
2089	829		NOT USED
2099	833	11	
2100	834		P1RAVG
2101	835	2	
2102	836		D1RAVG
2103	837	2	
2104	838		X1RAVG
2105	839	2	
2106	83A		P2RAVG
2107	83B	2	
2108	83C		D2RAVG
2109	83D	2	
2110	83E		X2RAVG
2111	83F	2	
2112	840		STACK BOTTOM
2385	951	274	STACK INITIALIZATION
2386	952		NOT USED
4095	FFF	1710	

A.

Source Listings of Program Algorithms

Creating Programs on the Cromemco System Three Microcomputer

The main program, five subroutines (table 3), and two interrupt service routines (table 3) that comprise the programming for the energy monitoring/display system were all developed on a Cromemco System Three microcomputer. The basic algorithms to accomplish the tasks given under "Program Development--Objectives" were worked out in the form of a main program and various subroutines and service routines and were entered as files on the microcomputer. The subroutine ADDREG, which adds a triple precision number in memory to a triple precision number that has its first two bytes in registers C and D, respectively, is used as an example here (table 4) and later on in the report (table 5) to show how the programming presented in this report was created.

The lines shown at the top of the example are first printed after the computer has been "booted up," which is done by depressing the return key a few times. The sequence involved in creating this file entitled "ADDREG.Z80" is shown in table 4, where all underlined characters are those entered by the programmer. EDIT is called first, and the program is entered by the programmer. EDIT is then exited, whereupon the file ADDREG.Z80 is created by the computer. Finally, the "TYPE" command is used to print out the file to verify that it is correct. A thorough account of the operating system and the creation of programs on the Cromemco System Three can be found in references 9 and 10, respectively.

Six-Channel Energy Monitoring/Display System Program Description and Usage

Introduction. - The programming for the energy monitoring/display system will now be discussed in detail. The main program consists of various segments that perform specific tasks. Each segment is set apart from the rest by a comment statement that serves as a heading (refer to the main program source listing, table C1), and in the following discussion each segment will be referenced by its heading. In addition, the flow diagram

Table 3. List of subroutines and service routines.

SUBROUTINES

Name	Function
REARRG	Takes the two bytes from the A-to-D containing the 11 data bits of a given channel's energy for a firing and rearranges them into the correct two-byte binary number.
ADDREG	Adds two triple precision numbers.
DIVIDE	Performs a 32-bit by 16-bit unsigned divide.
SWSEL1	Determines which channel is to be displayed on the display being considered, converts the energy average to BCD, then programs the thousand's and hundred's digits of the display.
SWSEL2	Programs the ten's and one's digits of the display under consideration with the energy data for the channel selected in subroutine SWSEL1.

SERVICE ROUTINES

DATRNSFR	Handles the transfer of data from the Z80 to the PDP-11 with handshaking.
PDPSER	Upon an $\overline{\text{NMI}}$ interrupt by the PDP-11, inputs data bits PA0-PA3 from the PDP-11 and checks if all of these bits are set. If they are, the PDP-11 has not changed the value of N but requires the current running average for each channel. Otherwise, bits PA0-PA3 represent N and bit 4 is set to indicate that the PDP-11, as opposed to the switches, has provided N.

Table 4. Example illustrating program creation.

```
@DOS version 02.17
Cromemco Disk Operating System
Copyright (c) 1978, 1979 Cromemco, Inc.

A.
A.EDIT ADDREG.Z80

CROMEMCO Text Editor version 00.10

New File
*I
ADDREG: LD      A,C
        ADD     A,(HL)
        LD      (HL),A
        INC     HL
        LD      A,D
        ADC     A,(HL)
        LD      (HL),A
        INC     HL
        LD      A,0
        ADC     A,(HL)
        LD      (HL),A
        RET
*E
Goodbye
End of Input File

A.TYPE ADDREG.Z80
ADDREG: LD      A,C
        ADD     A,(HL)
        LD      (HL),A
        INC     HL
        LD      A,D
        ADC     A,(HL)
        LD      (HL),A
        INC     HL
        LD      A,0
        ADC     A,(HL)
        LD      (HL),A
        RET

A.
```

Table 5. Example illustrating program conversion from source listing mnemonics into machine language (assembled).

```
A.ASMB ADDREG HEX=120
CROMEMCO CDOS Z80 ASSEMBLER version 02.15

Errors              0

end of assembly

A.TYPE ADDREG.PRN


CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

0120  79          0001 ADDREG: LD      A,C
0121  86          0002         ADD     A,(HL)
0122  77          0003         LD      (HL),A
0123  23          0004         INC     HL
0124  7A          0005         LD      A,D
0125  8E          0006         ADC     A,(HL)
0126  77          0007         LD      (HL),A
0127  23          0008         INC     HL
0128  3E00        0009         LD      A,0
012A  8E          0010         ADC     A,(HL)
012B  77          0011         LD      (HL),A
012C  C9          0012         RET

Errors              0

A.
```

symbols (the blocks and diamonds) of the general flow diagram (fig. B1) correspond to the various segments of the main program and are labeled with lower case letters so that the general flow diagram can be used along with the source listing and flow chart of the main program in the discussion of each of the segments of the main program which now follows.

Initialize stack and set interrupt mode. - After the energy-averaging switches and the display-selection switches have been set, and after the laser system has been activated by the operator, the Z80 energy monitoring/display program is executed. To start, the interrupt enable flip-flop is set so that maskable interrupts can be handled, since there is a service routine for maskable ($\overline{INT}$) as well as nonmaskable ($\overline{NMI}$) interrupts. Maskable and nonmaskable interrupts are the means by which the PDP-11 can direct the operation of the Z80 microprocessor. How this is accomplished will be discussed in more detail when each of the interrupt service routines is described.

The Z80 microprocessor can handle three different types of $\overline{INT}$ interrupt modes. Mode one is chosen since there is only one maskable interrupting source, the PDP-11/34 minicomputer* (fig. F4). In this mode, a maskable interrupt causes the contents of the program counter to be saved in the stack. A restart is then made beginning at address 38H, where a jump instruction to the beginning of service routine DATRNSFR is located (see table 1).

Because the program utilizes stack operations, a section of memory (840H - 950H) is designated as a memory stack in order to facilitate the handling of data and information. The stack is used to hold return addresses for subroutines and to hold the contents of registers that need to be freed temporarily for use within the main program or subroutines.

Read switches and arrange into new N. - The three data bits corresponding to the settings of the energy-averaging switches are input into the accumulator from port 0FH (see figs. F2 and F6). Data bits AV0, AV1, and AV2 correspond to the switch settings and are located in bits three (D3),

---

*The PDP-11 is also configured for nonmaskable interrupts ($\overline{NMI}$), as shown in figure F4.

four (D4), and five (D5) of the accumulator, respectively. After being input, the data from this port appears in the accumulator as shown in figure 1.

D7	D6	D5	D4	D3	D2	D1	D0
X	X	AV2	AV1	AV0	X	X	X

Figure 1. Accumulator contents after input of data from port OFH. (The X's denote irrelevant data bits.)

Bits two, six, and seven of the accumulator are reset, and the contents of the accumulator are then shifted twice to the right (fig. 2).

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	AV2	AV1	AV0	0

Figure 2. Accumulator contents after shifting twice right.

In this form, the accumulator's contents correspond to N, with 2^N representing the number of transmissions to be averaged. Bit 4 (D4) is reset to indicate that the energy-averaging switches (rather than the PDP-11) have provided N. The accumulator's contents are then stored at the address labeled NADDR (see Appendix A). Alternatively, N can be provided in software by a directive from the PDP-11. In this case, data bits PA0 to PA3 (figs. F4 and F6), after being input into the accumulator (bits D0 - D3, respectively), correspond to N. Bit four of the accumulator is set to indicate that, until further directed, N as provided by the PDP-11 is to be used and that on future runs the energy-averaging switches should not be read. The PDP-11 can change N by generating a nonmaskable ($\overline{\text{NMI}}$) interrupt, which is actually the "NEW DATA READY" pulsed signal from the DR11-C interface module (figs. F4 and F6). This causes the interrupt service routine PDPSER to be entered.

Within service routine PDPSER, data bits PA0 to PA3 are input into the accumulator from the PDP-11 through port OEH, where they occupy the first four bit positions, D0 to D3, respectively. All of the higher order bits

are reset. IF the accumulator's contents are OFH, the PDP-11 has not changed the value of N but requires the current running average for each channel. Otherwise, bits 0 to 3 of the accumulator represent N and bit four is set to indicate that the PDP-11 (as opposed to the energy-averaging switches) has provided N. The accumulator's contents are then stored in memory at the address labeled "NADDR." Table 6 shows the values of N that correspond to the allowable arrangements of AV0 to AV2 and PA0 to PA3. Note that the switches, through AV0 to AV3, can specify N = 0, 2, 4, 6, 8, 10, 12, and 14 only, while the PDP-11 can specify N = 0 to 14 through bits PA0 to PA3.

Calculate 2^N for # firings to be averaged. - The desired number of firings to be averaged is determined from N by simply calculating 2^N. To do this the number stored in memory at NADDR is compared to values of 0 through 14 in turn until a true comparison is made. The value of N is then known to the program and 2^N is specified by setting the bit of the two-byte number (labeled "NUMFIR"; see Appendix A) that is the same as N. For example, if N is nine, then bit nine of the two-byte number at NUMFIR is set with all the remaining bits reset.

Clear memories holding channel energy sums. - The 18 memory locations in RAM that hold the 3-byte sums of the energies for the six channels need to be initialized to zero before each averaging run. These running sums are used to calculate the running average for each channel after each firing of the laser system. They are initialized by loading each of the 18 memory locations with a zero beginning with address P1ESUM and ending with address X2ESM3 (see table 2 and Appendix A).

Initialize shot counter. - The shot counter (register pair BC) is initialized to zero before each averaging run and is incremented after each firing of the laser system in order to keep track of the number of firings. In addition to keeping track of this number, the shot counter is used in the calculation of the running averages after each firing.

Check if laser system has fired. - In order to determine when the laser system has fired, the A-to-D converter busy signals for channels zero and three (B0 and B3) corresponding to pump lasers P1 and P2, respectively, are monitored. When one of the pump lasers has fired, laser P1 say, its

Table 6. Data bit configurations that correspond to allowed values of N.

D3 AV2, PA3	D2 AV1, PA2	D1 AV0, PA1	D0 0, PA0	N
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14

busy signal (B0) goes high for about 40 ms, which is the conversion time. At the end of this time the digitized value for the energy of the pulse just fired is on data lines DB0 to DB11.

To determine when a firing has occurred, the busy signal for laser P1 is input from port 06H (see figs. F1 and F6). If this signal is high, a firing has occurred and this signal (B0) is subsequently monitored to find out when it goes low again, at which point valid data for the energy of the transmission is on data lines DB0 to DB11. If the busy signal for laser P1 is not high at the time it is examined, then the busy signal for laser P2 (B3) is examined to see if it is high. If so, the laser system has fired and this signal (B3) is monitored to see when it goes low, at which point valid data is on the data lines. If neither busy signal (B0 or B3), as given by the data from port 06H, is high when examined, then the program loops back and new busy signals (B0 and B3) are input from port 06H and the procedure outlined above is resumed.*

Once it has been determined that the laser system has fired, the next task is to input the energies of each of the six channels as provided by their respective A-to-D converters.

Sum channel energies to previous totals. - The energy transmitted during the firing for each of the six channels is summed to the previous total in memory. The low byte of the energy for channel zero is retrieved from the channel zero A-to-D converter via port 00H, and the high byte is input from port 08H (fig. F1). Table 7 lists the energy data input ports:

Table 7. Energy data input ports.

CHANNEL	LOW BYTE	HIGH BYTE
0	port 00H	port 08H
1	port 01H	port 09H
2	port 02H	port 0AH
3	port 03H	port 0BH
4	port 04H	port 0CH
5	port 05H	port 0DH

*Extra hardware could have been used together with an interrupt for this function. Software was chosen as a less expensive and more versatile method.

The two bytes of a given channel's energy are not in the optimum format after being input; they come in as shown in figure 3.

	7	6	5	4	3	2	1	0
LOW BYTE (register E)	X	X	DB5	DB4	DB3	DB2	DB1	DB0

	15	14	13	12	11	10	9	8
HIGH BYTE (register D)	X	X	DB11	DB10	DB9	DB8	DB7	DB6

Figure 3. Format of raw energy data from A-to-D converters. (The X's denote irrelevant data.)

Subroutine REARRG takes the two bytes shown in figure 3 and rearranges them into the correct two-byte binary number that corresponds to the energy of the transmission for the given channel. The resulting two-byte binary number appears as shown in figure 4.

	7	6	5	4	3	2	1	0
LOW BYTE	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

	15	14	13	12	11	10	9	8
HIGH BYTE	0	0	0	0	DB11	DB10	DB9	DB8

Figure 4. Format of energy data after subroutine REARRG.

In this form, the energy can be summed to the previous total for the channel, and this is done by subroutine ADDREG.

Advance shot counter. - The shot counter (register pair BC) is incremented by one after each firing of the laser system to keep account of the number of firings during a given averaging run. As previously mentioned, the value of the shot counter is also used in the calculation of the running averages, which is the next task of the program.

Calculate running averages for each channel. - The running or current average for a given channel's energy per transmission is calculated by dividing the running sum for the channel by the contents of the shot counter (register pair BC). This is done for each of the six channels and the two-byte result is stored in memory. For example, the two-byte result for channel zero (laser P1) is stored in RAM beginning at address P1RAVG (see table 2 and Appendix A). Subroutine DIVIDE, which does the actual division of the two binary numbers when calculating a running average, can be found on page 235 of The Z80 Microcomputer Handbook by William Barden, Jr. (ref. 7). This subroutine divides the 32-bit number found in registers H, L, D, and E by the 16-bit number in register pair BC. Since the running sum for any given channel's energy is, at most, 24 bits long, the high byte in register H is always made equal to zero before this subroutine is called.

Check if N firings reached. - The value of the shot counter (register pair BC) is compared to the value of N stored in memory. If there is a true comparison, the next segment (k) of the program is begun as the desired number of transmissions to be averaged has been achieved. Otherwise, the program loops back to segment f to wait for the laser system to fire once more.

Send low byte of P1 energy average to PDP-11. - The low byte of the energy average per transmission for channel zero (laser P1) is output to port 00H, which is one of the two eight-bit parallel output latches connected to the DR11-C interface module (figs. F4 and F6). In addressing this latch, the "REQUEST A" line of the DR11-C interface becomes asserted. This signals the PDP-11 that energy average data is ready to be transferred to it.

Wait 12 μs for PDP-11 response. - A 12-μs period is used to wait for the PDP-11 response. If the PDP-11 does not respond within this time period, the programming of the three-channel selectable display is begun. If the PDP-11 does respond in time, the response is made in the form of a maskable interrupt ($\overline{INT}$) generated by a strobe from the DR11-C, then the remainder of the energy average data is transferred to the PDP-11.

Transfer of energy average data to PDP-11. - The high byte of the energy average for channel zero (laser P1) is output to port 01H, which is

the other eight-bit parallel output latch (figs. F4 and F6). It is important to note that if the PDP-11 inputs the first byte of energy average data, that is, if it responds within 12-μs after the low byte for channel zero has been transferred, it is assumed that the PDP-11 will respond to the rest of the 11 transfers of data bytes that correspond to the 2-byte averages for each of the 6 channels.

In each transfer of a single byte of energy average data, except for the very first, the low byte of a given channel's average is output to port 00H, after which the program loops indefinitely until interrupted by a maskable interrupt ($\overline{\text{INT}}$; strobe signal from DR11-C). The high byte is then output to port 01H, and the strobe signal that will interrupt the looping process is again awaited. In this way all 12 bytes of the energy averages are transferred to the PDP-11. It is important that after accepting the first data byte the PDP-11 input the remainder of the data bytes when they are put on the data lines. The transfer sequence of the data bytes to the PDP-11 is given in table 8.

The maskable interrupts, which comprise the handshaking between the Z80 and the PDP-11, cause the service routine DATRNSFR to be entered. The sole purpose of this service routine is to determine the address of the instruction in the main program that will cause the next byte of energy average data to be transferred and to return to that point in the main program.

Program three-channel selectable display. - The three-channel selectable display can display the energy averages of any three of the six channels. For each of the three displays, three data bits are input for use in determining which channel has been selected for display (figs. F3 and F6). For the left-hand display, the data bits are S0, S1, and S2, which are input from port 07H. For the middle display, the data bits are S3, S4, and S5, which are also input from port 07H. The data bits for the right-hand display are S6, S7, and S8, and these are input from port 0FH. How a channel to be displayed is determined from the set of three data bits is shown by table 9.

Table 8. Sequence of data transfer from Z80 to PDP-11.

DATA BYTE	PORT OUTPUT TO	CONDITION FOR NEXT TRANSFER
Low Byte of P1 Avg.	00H	PDP-11 inputs within 12 μs
High Byte of P1 Avg.	01H	($\overline{INT}$) via strobe
Low Byte of D1 Avg.	00H	($\overline{INT}$) via strobe
High Byte of D1 Avg.	01H	($\overline{INT}$) via strobe
Low Byte of X1 Avg.	00H	($\overline{INT}$) via strobe
High Byte of X1 Avg.	01H	($\overline{INT}$) via strobe
Low Byte of P2 Avg.	00H	($\overline{INT}$) via strobe
High Byte of P2 Avg.	01H	($\overline{INT}$) via strobe
Low Byte of D2 Avg.	00H	($\overline{INT}$) via strobe
High Byte of D2 Avg.	01H	($\overline{INT}$) via strobe
Lowe Byte of X2 Avg.	00H	($\overline{INT}$) via strobe
High Byte of X2 Avg.	01H	($\overline{INT}$) via strobe

Table 9. Switch selections for three-channel display.

S0, S1, S2 or S3, S4, S5 or S6, S7, S8	DATA BYTE VALUE	CHANNEL SELECTED FOR DISPLAY
0 0 0	00H	P1
0 0 1	01H	D1
0 1 0	02H	X1
0 1 1	03H	P2
1 0 0	04H	D2
1 0 1	05H	X2

After the data bits for a given display have been input, they are arranged into a data byte in which the three data bits comprise bits 0 to 2 with the remainder of the bits of the data byte all reset. Subroutine SWSEL1 is then called. The first part of this subroutine compares the data byte, in turn, to the six data byte values (table 9). In this way, the channel that has been selected for display is determined.

Subroutine SWSEL1 next converts the energy average for the channel selected into four binary-coded decimal (BCD) numbers corresponding to a one's, a ten's, a hundred's, and a thousand's digit. These BCD numbers are used to program the seven segment displays. This is done by arranging the highest four bits of register D so that they match the BCD number corresponding to the thousand's digit. The lower four bits are made to match the BCD number corresponding to the hundred's digit. When SWSEL1 is exited, the seven segment displays corresponding to these digits are programmed by putting the contents of register D on the data lines and outputting them to the appropriate port (2, 3, 4, 5, 6, or 7). A similar operation is done by subroutine SWSEL2, which programs the seven segment displays corresponding to the ten's and one's digits. In this way the three-channel selectable display is programmed to display any three of the six energy averages.

Determine where to resume main program. - Bit four of the byte for N at address NADDR is tested. If it is reset, then the energy-averaging switches

are to be read for N and the program resumes at that point (segment b). If bit four is set, the energy-averaging switches are not read, since N has been provided earlier by the PDP-11 through an $\overline{NMI}$ interrupt, and the program resumes at the point where 2^N, the number of firings for the next averaging run, is calculated (segment c).

Conversion of Assembly Language Program into Machine Language

Once a file has been created for a program, it needs to be converted into machine code. This is done on the Cromemco microcomputer by its macroassembler. The listing in table 5 demonstrates the sequence a programmer would use to assemble the source program ADDREG.Z80. The HEX = 120 option specifies the run address for ADDREG.Z80 as it is to run out of ROM. In using this option, the address of the first byte of instruction will be 120H, as is shown on the assembled listing (table 5). The instruction "TYPE ADDREG.PRN" causes the whole assembled listing to be printed out. Further information regarding the assembly of programs on the Cromemco System Three Microcomputer can be found in the Cromemco macroassembler manual (ref. 11).

Program Verification

An assembled HEX file can be tested on the Cromemco System Three microcomputer by using DEBUG. When DEBUG is called, files can be read into memory at the addresses shown in their assembled listings. In the example following (table 10), both MAIN.Z80 and ADDREG.Z80 have been assembled as HEX files and can be read into memory using DEBUG. To set up the portions of the programs being tested here, addresses 800H to 802H are loaded with the values 11H, 11H, and 01H, respectively. Register pair DE is loaded with the value 0FFFH. The program counter is set to the point in the main program where the number in registers B and C is added to the energy sum for laser P1 (address 4F4H). The addition is performed by subroutine ADDREG.

The printout of table 10 shows how each instruction can be stepped through. After the return from ADDREG, which is called by the main program in the sequence being tested here, the result of the addition can be found at the addresses for the energy sum for laser P1 (addresses 800H - 802H) and is what is printed out on the last line of the example, that is, 11111H plus

Table 10. Example illustrating assembled hexadecimal file testing using DEBUG.

```
A.DEBUG
DEBUG version 00.08
-FMAIN.HEX
-R
NEXT = 0674
-FADDREG.HEX
-R
NEXT = 0674
-SM800
0800 0800' BB  11
0801 0801' CD  11
0802 0802' 01  .
-SD
DE=0000 0FFF
-SP
P=0100 0100' 4F4
-DR
        A=00 BC=0000 DE=0FFF HL=0000 S=0100 P=04F4 04F4' LD    HL,0800
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00        (0800')
-T
        A=00 BC=0000 DE=0FFF HL=0800 S=0100 P=04F7 04F7' LD    C,E
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
        A=00 BC=00FF DE=0FFF HL=0800 S=0100 P=04F8 04F8' CALL  0120
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00        (0120')
-T
        A=00 BC=00FF DE=0FFF HL=0800 S=00FE P=0120 0120' LD    A,C
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
        A=FF BC=00FF DE=0FFF HL=0800 S=00FE P=0121 0121' ADD   A,(HL)
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H  C  A=10 BC=00FF DE=0FFF HL=0800 S=00FE P=0122 0122' LD    (HL),A
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H  C  A=10 BC=00FF DE=0FFF HL=0800 S=00FE P=0123 0123' INC   HL
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H  C  A=10 BC=00FF DE=0FFF HL=0801 S=00FE P=0124 0124' LD    A,D
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H  C  A=0F BC=00FF DE=0FFF HL=0801 S=00FE P=0125 0125' ADC   A,(HL)
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H     A=21 BC=00FF DE=0FFF HL=0801 S=00FE P=0126 0126' LD    (HL),A
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H     A=21 BC=00FF DE=0FFF HL=0801 S=00FE P=0127 0127' INC   HL
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H     A=21 BC=00FF DE=0FFF HL=0802 S=00FE P=0128 0128' LD    A,00
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
  H     A=00 BC=00FF DE=0FFF HL=0802 S=00FE P=012A 012A' ADC   A,(HL)
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
        A=01 BC=00FF DE=0FFF HL=0802 S=00FE P=012B 012B' LD    (HL),A
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-T
        A=01 BC=00FF DE=0FFF HL=0802 S=00FE P=012C 012C' RET
       A'=00 B'=0000 D'=0000 H'=0000 X=0000 Y=0000 I=00
-DM800,S3
0800  10 21 01 -  .!.
-
```

OFFFH equals 12110H. A more thorough explanation of DEBUG can be found in reference 11.

ACKNOWLEDGMENTS

This effort was supportd by funds provided under NASA Cooperative Agreement NCC1-32 (Dr. E.V. Browell, Technical Monitor). The authors wish to acknowledge the assistanace of (1) David Livingston, graduate student, Electrical Engineering Department, Old Dominion University, who handled the preliminary logic design of the Z80-based microcomputer under the direction of Dr. M.R. Varanasi; (2) Woodrow W. Midgette, Jr., NASA-Electronics Packing Development Unit, who handled the layout engineering and packing design under the direction of T.K. Lusby; (3) William J. McCabe of NASA, who handled the debugging, calibration, and checkout; and (4) Jim Siviter and Loyd Overbay, who handled the power sensor head layout, fabrication, and installation.

The data included as Appendix E is provided by permission of Zilog, Inc., 10460 Bubb Road, Cupertino, CA 95104.

APPENDIX A

SYMBOLIC ADDRESSES

BINVAL	RAM address of two-byte binary number that is converted into its binary-coded decimal equivalent in subroutine SWSEL1
DECMAL	RAM address of binary-coded decimal number that corresponds to the one's digit; used in subroutine SWSEL1
DECML4	RAM address of binary-coded decimal number that corresponds to thousand's digit; used in subroutine SWSEL2
D1ESM3	RAM address of high byte of D1 energy sum; used in main program
D1ESUM	RAM address of low byte of channel D1 energy sum; used in main program
D1RAVG	RAM address of low byte of channel D1 running average; used in main program
D2ESM3	RAM address of high byte of channel D2 energy sum; used in main program
D2ESUM	RAM address of low byte of D2 energy sum; used in main program
D2RAVG	RAM address of low byte of channel D2 running average; used in main program and subroutine SWSEL1
HUNDRD	ROM address of low byte of binary equivalent of one hundred; used in subroutine SWSEL1
NADDR	RAM address of N; 2^N = the number of transmissions to be averaged; used in main program
NUMFIR	RAM address of low byte of number of transmissions to be averaged; used in main program
ONES	ROM address of low byte of binary equivalent of one; used in subroutine SWSEL1
P1ESM3	RAM address of high byte of channel P1 energy sum; used in main program
P1ESUM	RAM address of low byte of channel P1 energy sum; used in main program
P1RAVG	RAM address of low byte of channel P1 running average; used in main program and subroutine SWSEL1
P2ESM3	RAM address of high byte of channel P2 energy sum; used in main program

P2ESUM RAM address of low byte of channel P2 energy sum; used in main program

P2RAVG RAM address of low byte of channel P2 running average; used in main program and subroutine SWSEL1

TENS ROM address of low byte of binary equivalent of ten; used in subroutine SWSEL1

THOUSN ROM address of low byte of binary equivalent of one thousand; used in subroutine SWSEL1

X1ESM3 RAM address of high byte of channel X1 energy sum; used in main program

X1ESUM RAM address of low byte of channel X1 energy sum; used in main program

X1RAVG RAM address of low byte of channel X1 running average; used in main program and subroutine SWSEL1

X2ESM3 RAM address of high byte of channel X2 energy sum; used in main program

X2ESUM RAM address of low byte of channel X2 energy sum; used in main program

X2RAVG RAM address of low byte of channel X2 running average; used in main program and subroutine SWSEL1

APPENDIX B

GENERAL FLOW DIAGRAM

This Page Intentionally Left Blank

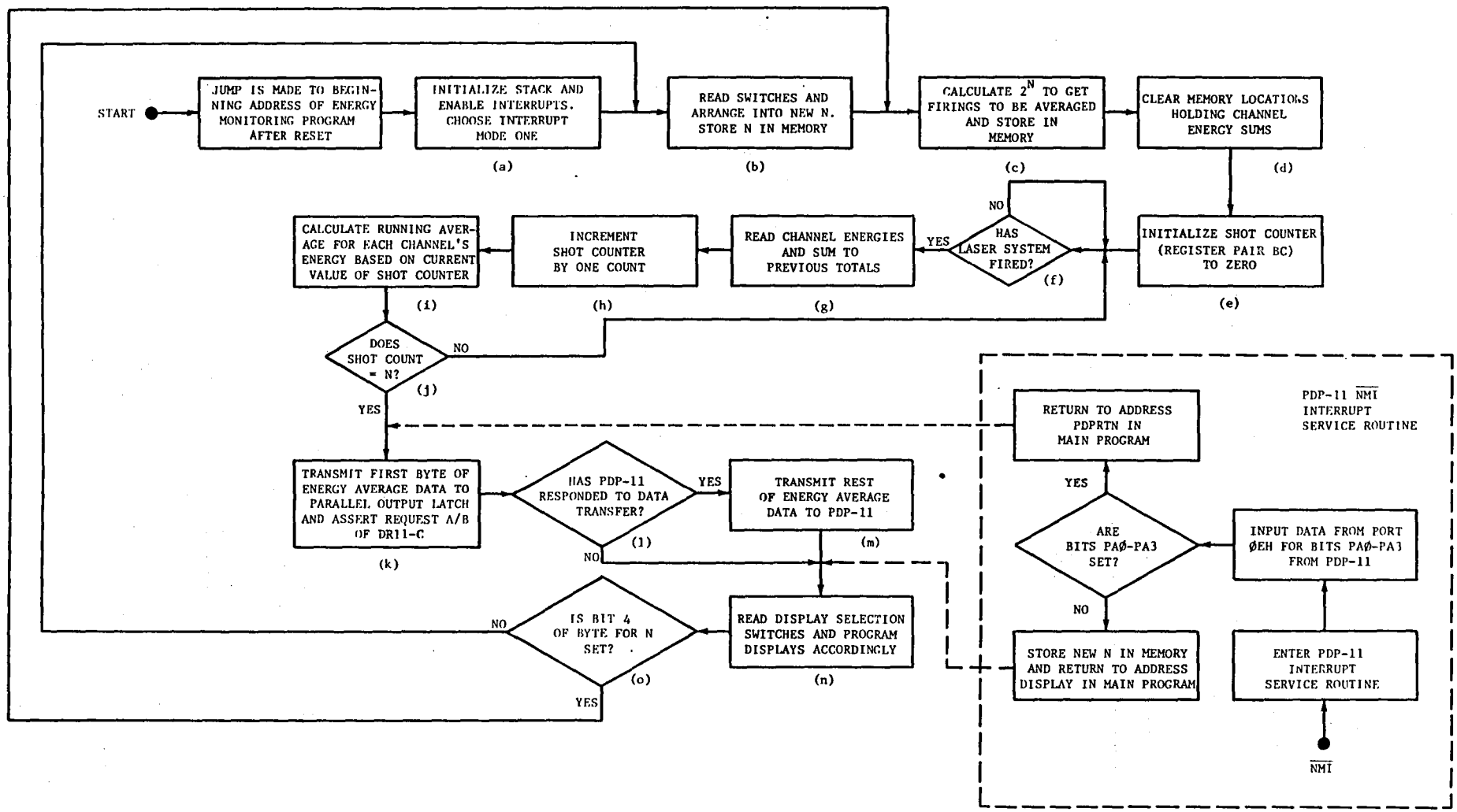


Figure B1. General flow diagram.

This Page Intentionally Left Blank

APPENDIX C

PROGRAM FLOW CHARTS AND SOURCE LISTINGS

This Page Intentionally Left Blank

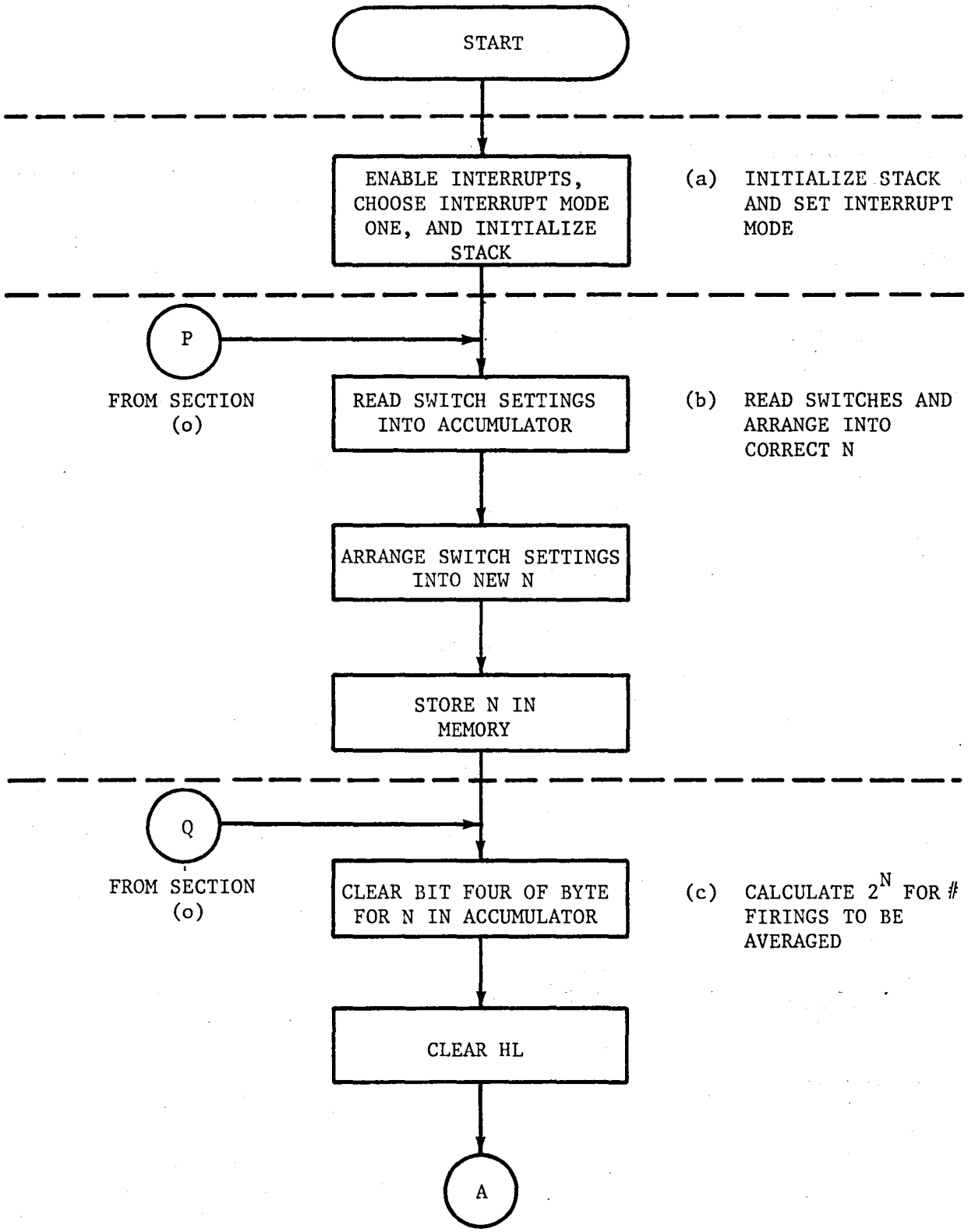


Figure C1. Flow chart for main program.

(c) CONTINUED

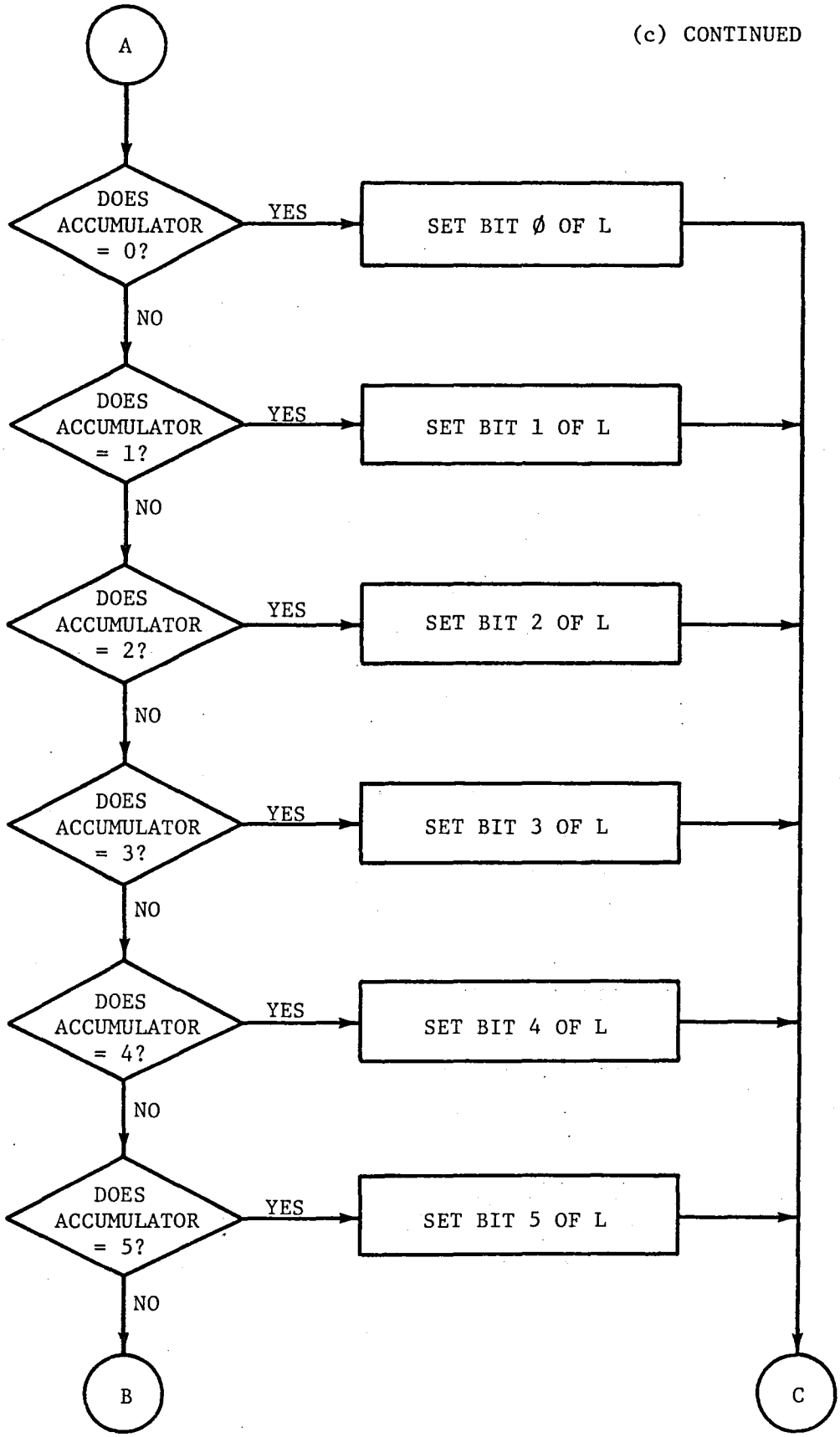


Figure C1. (continued).

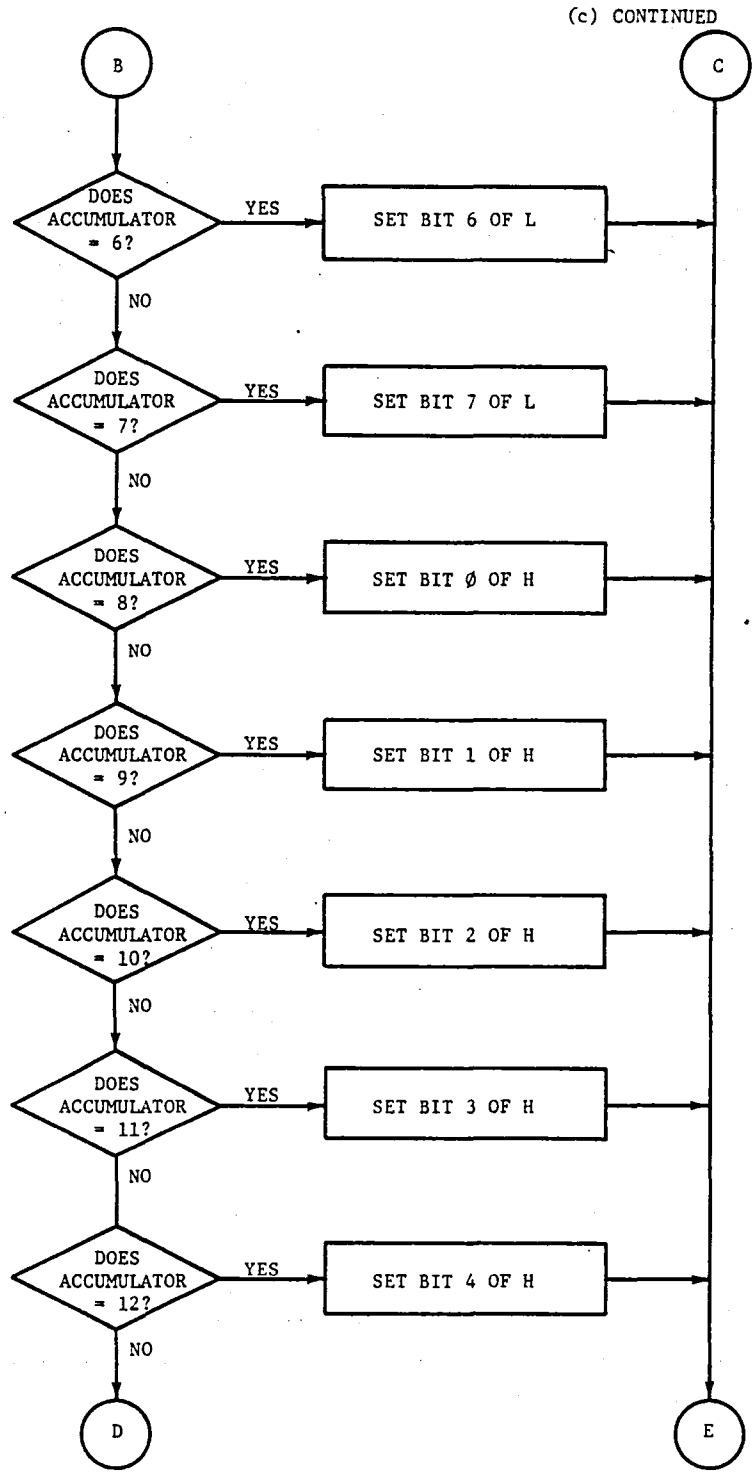


Figure C1. (continued).

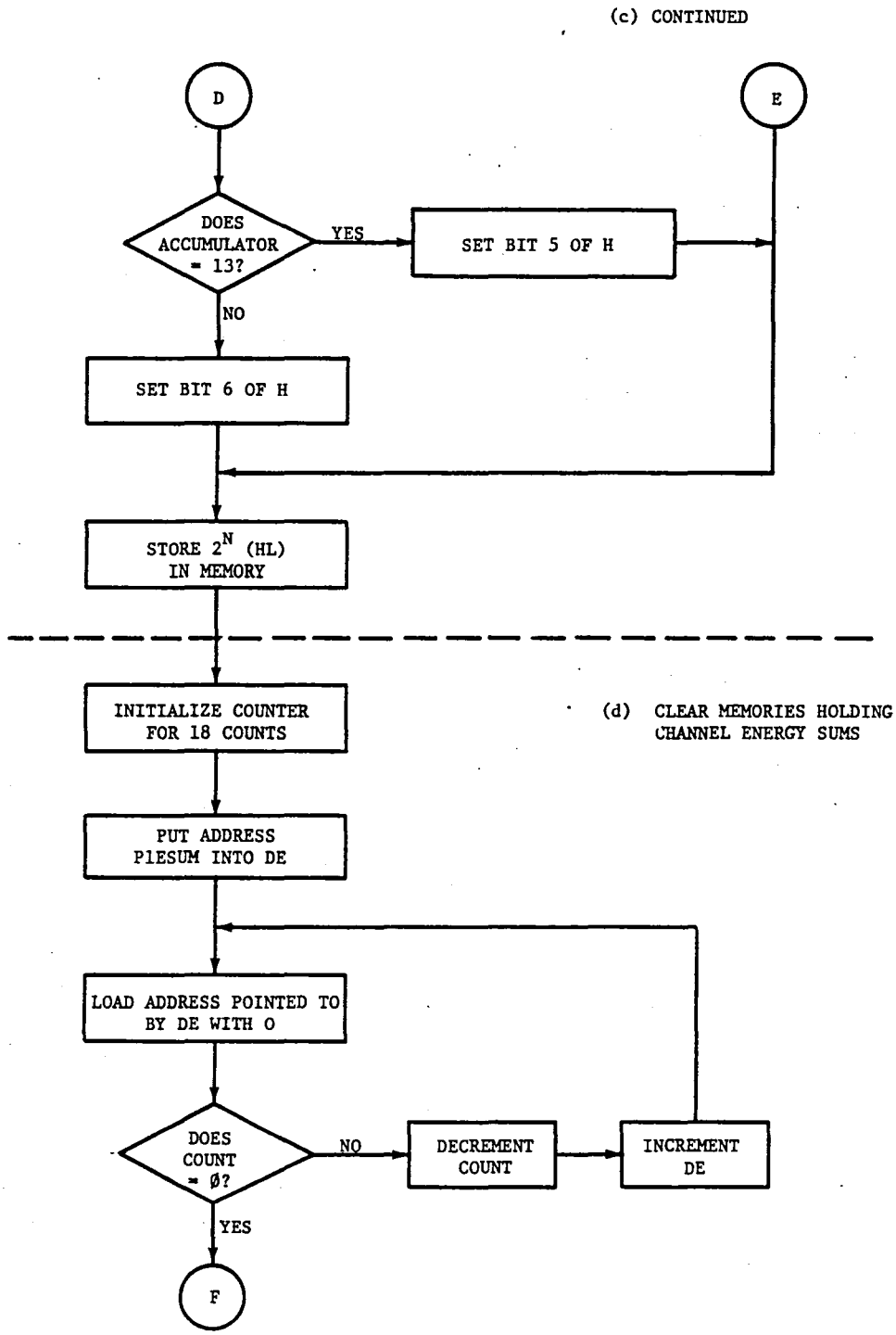


Figure C1. (continued).

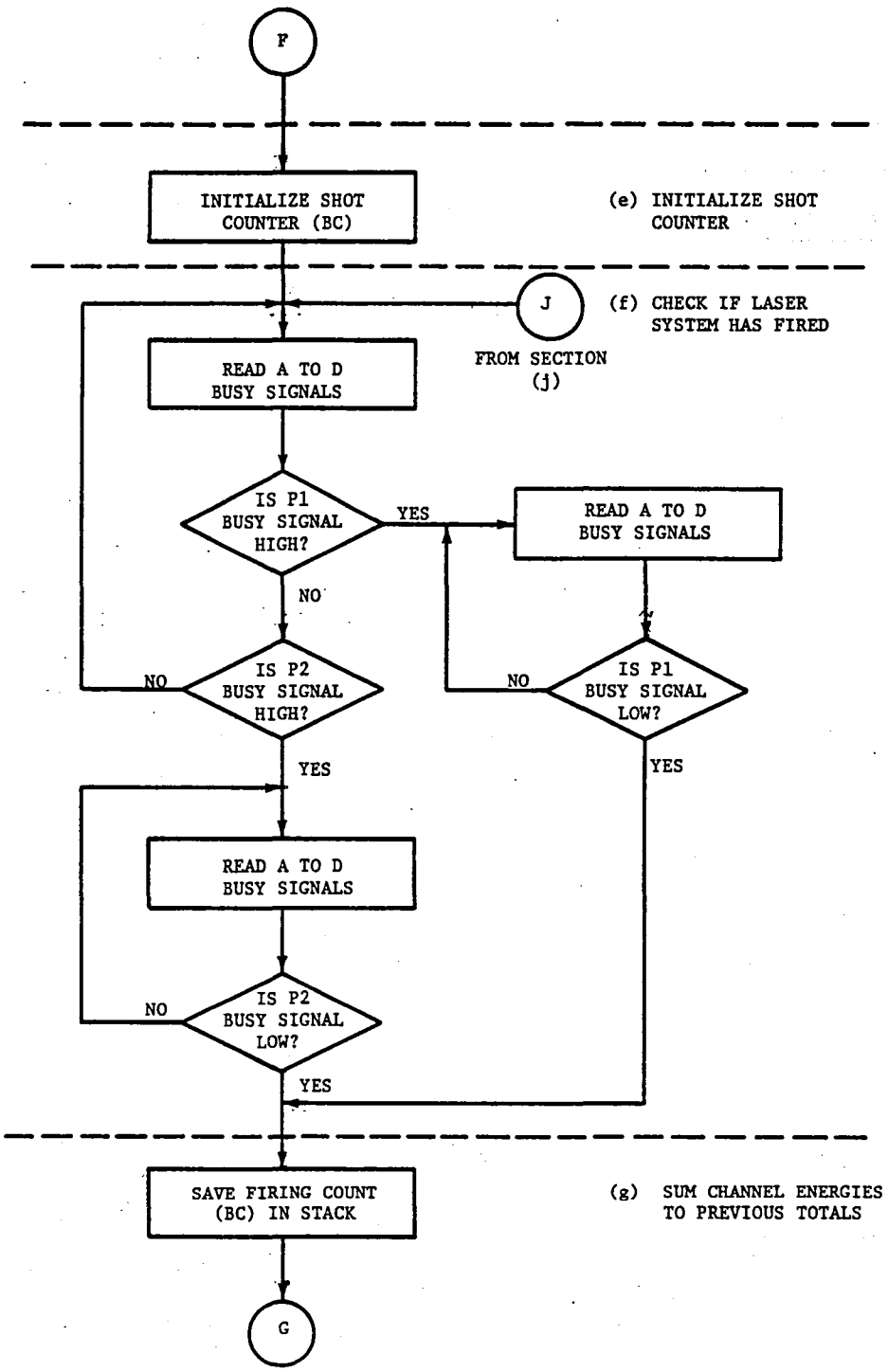


Figure C1. (continued).

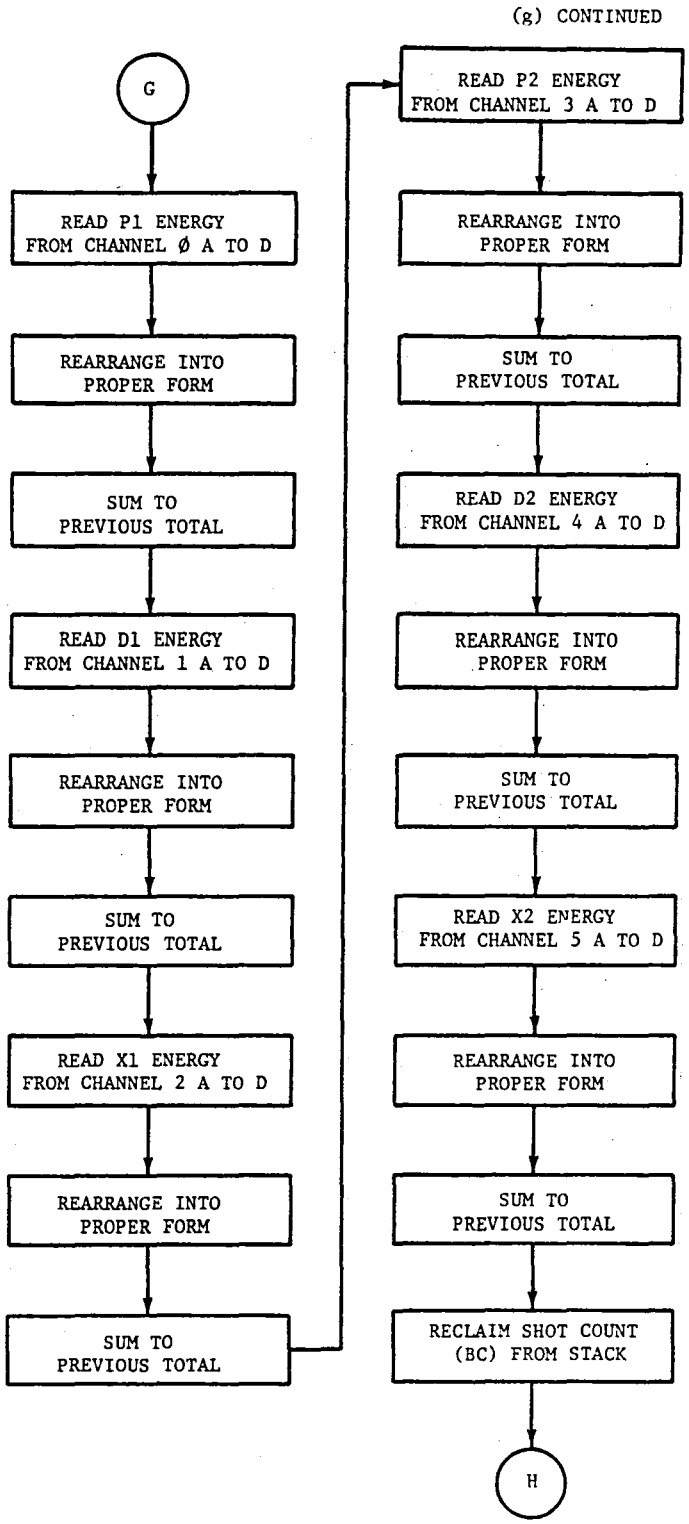


Figure C1. (continued).

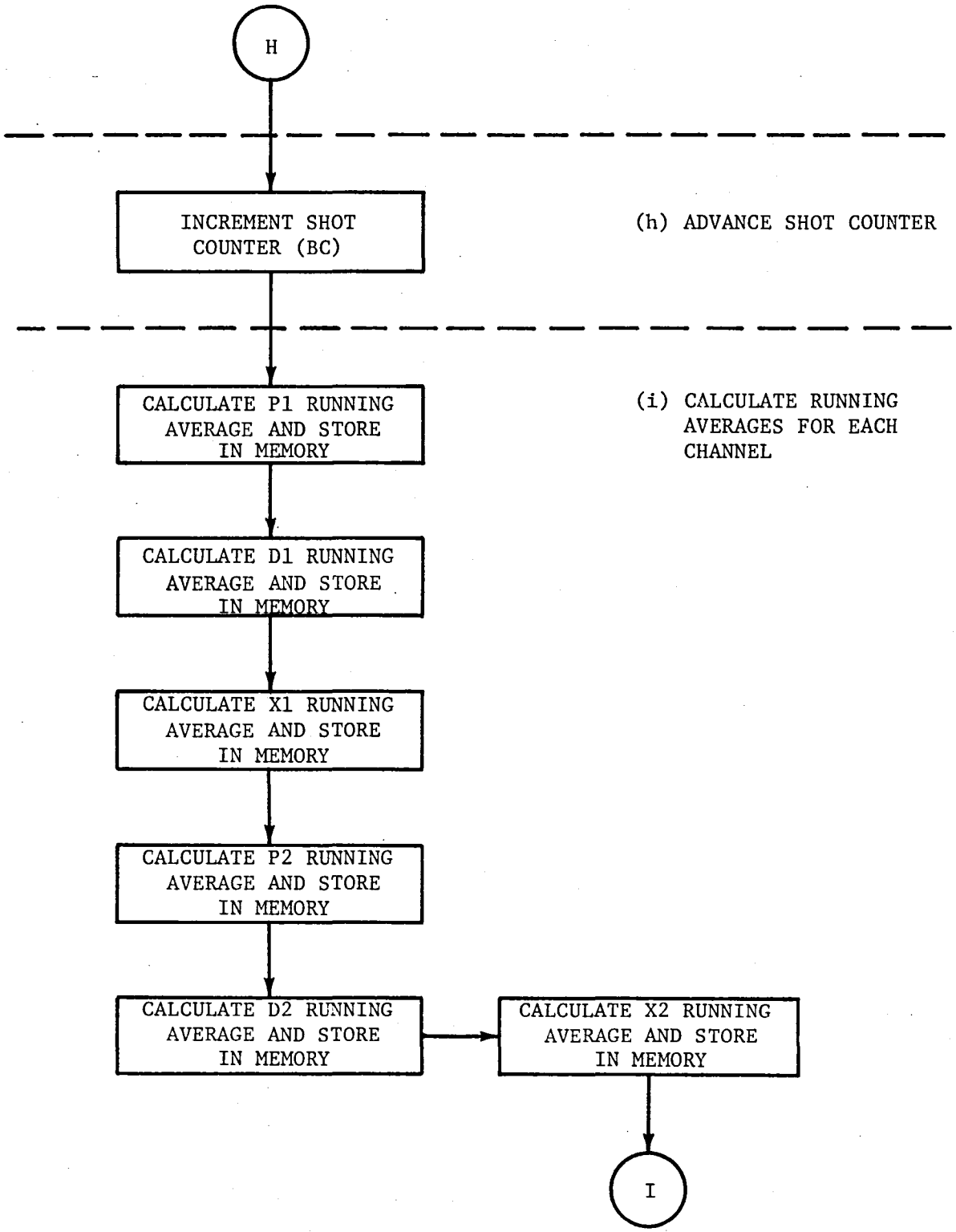


Figure C1. (continued).

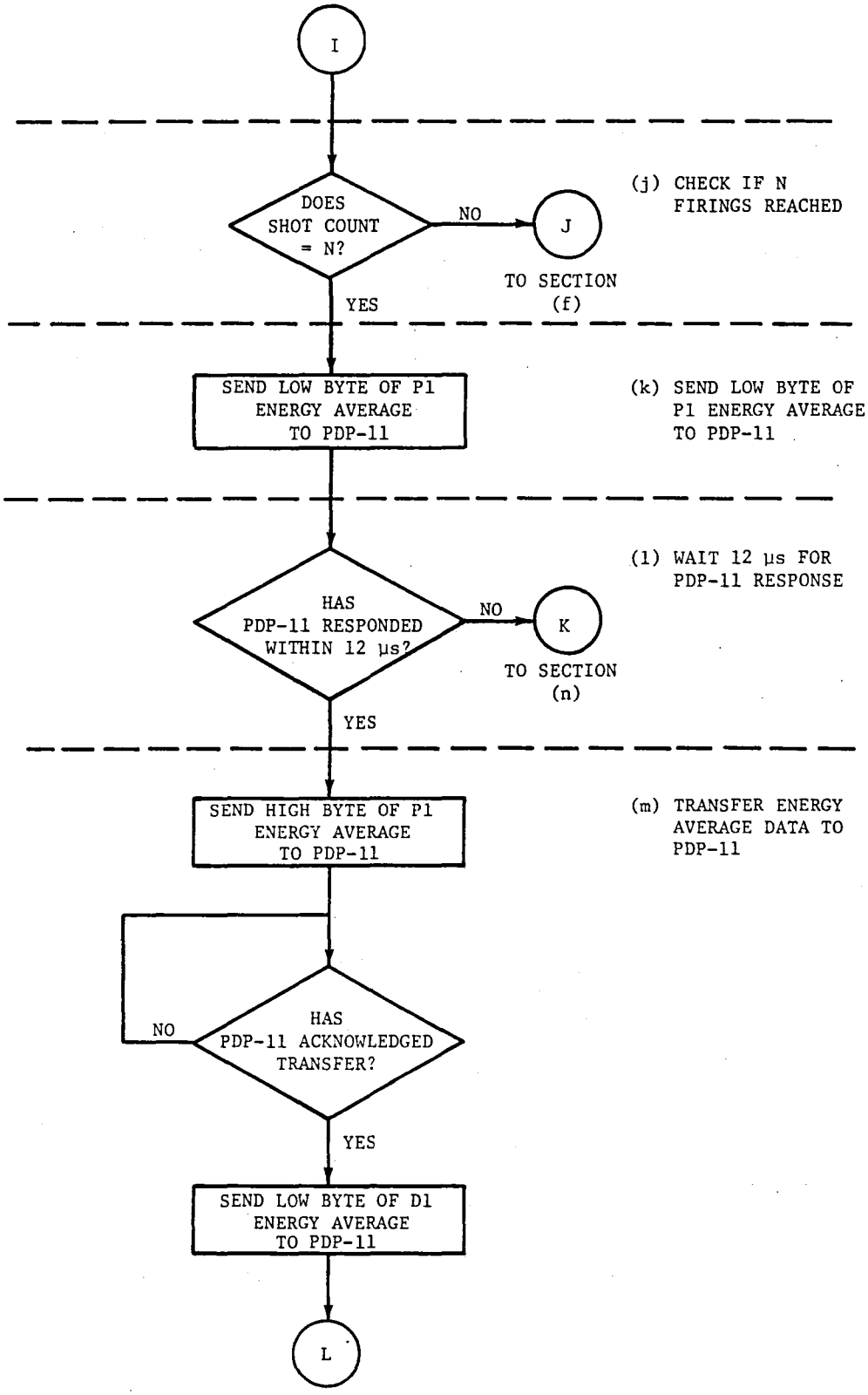


Figure C1. (continued).

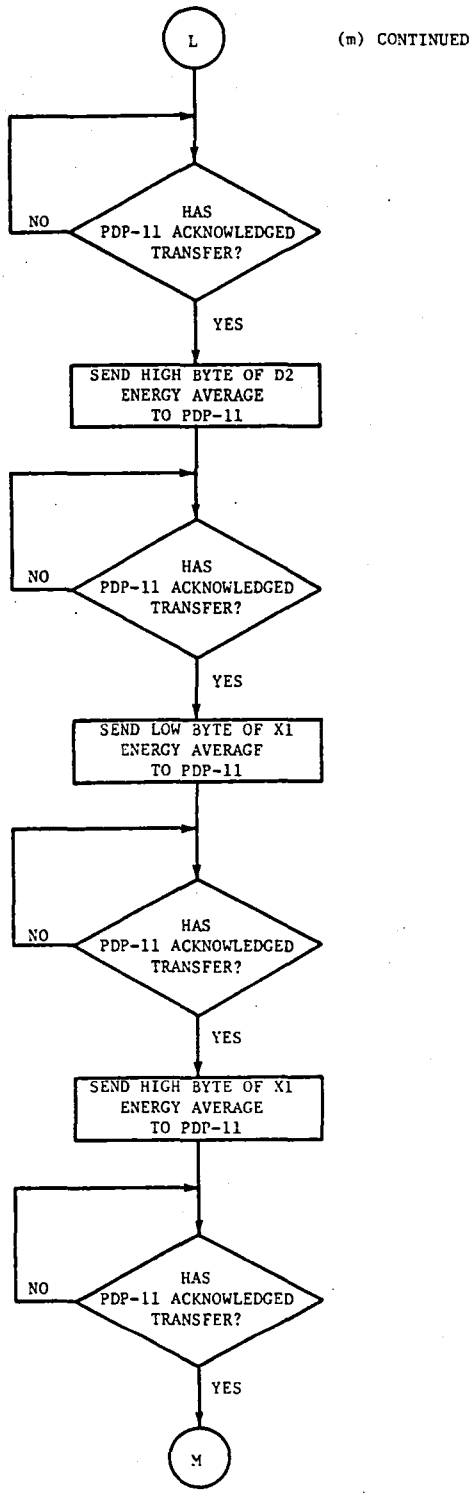


Figure C1. (continued).

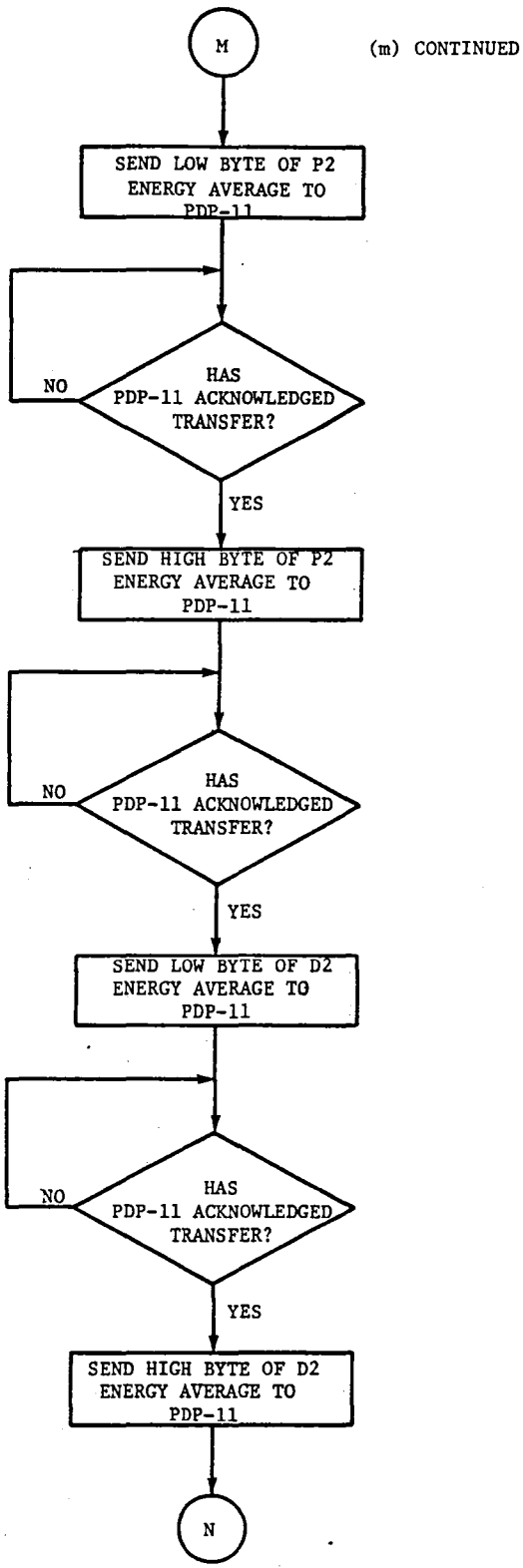


Figure C1. (continued).

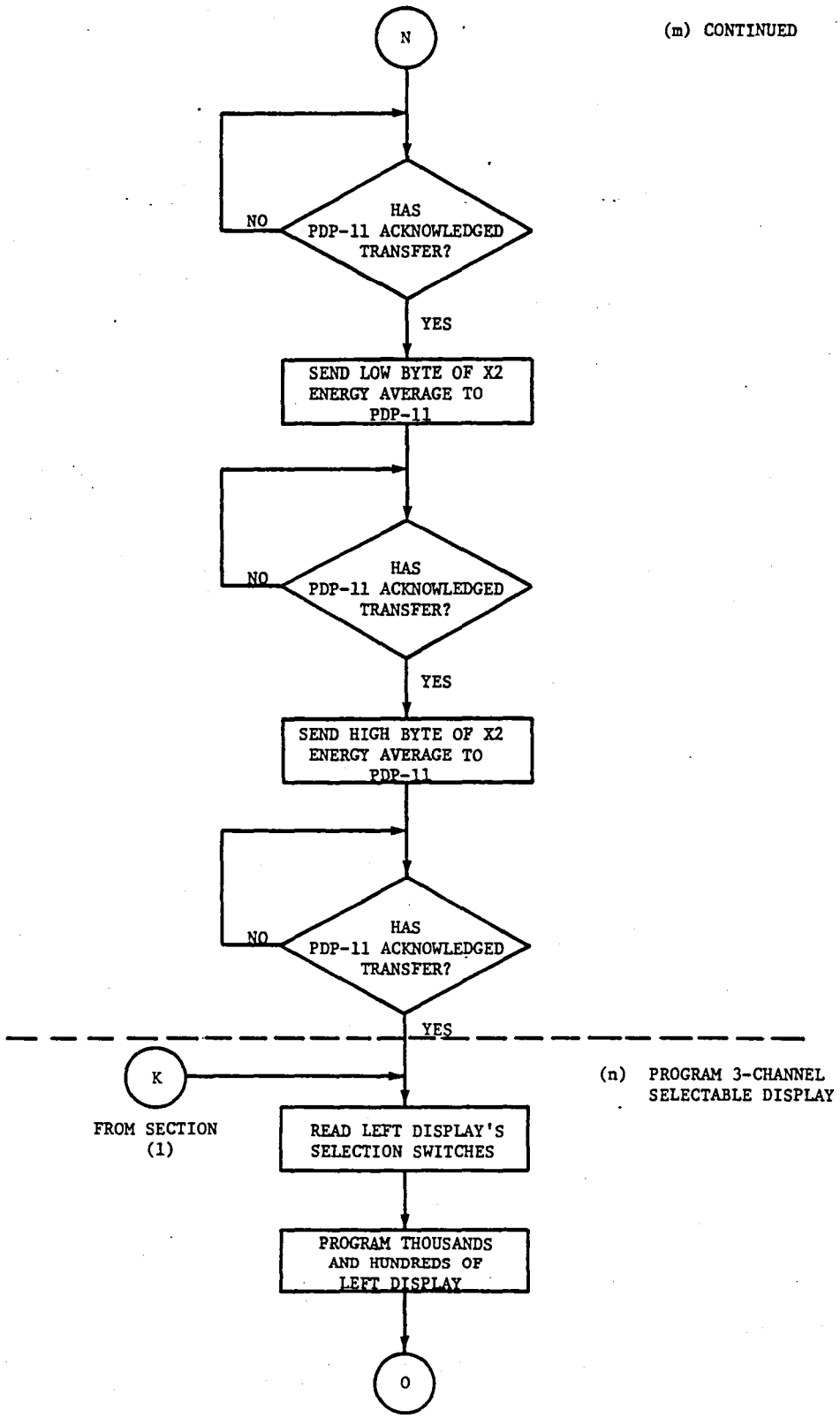


Figure C1. (continued).

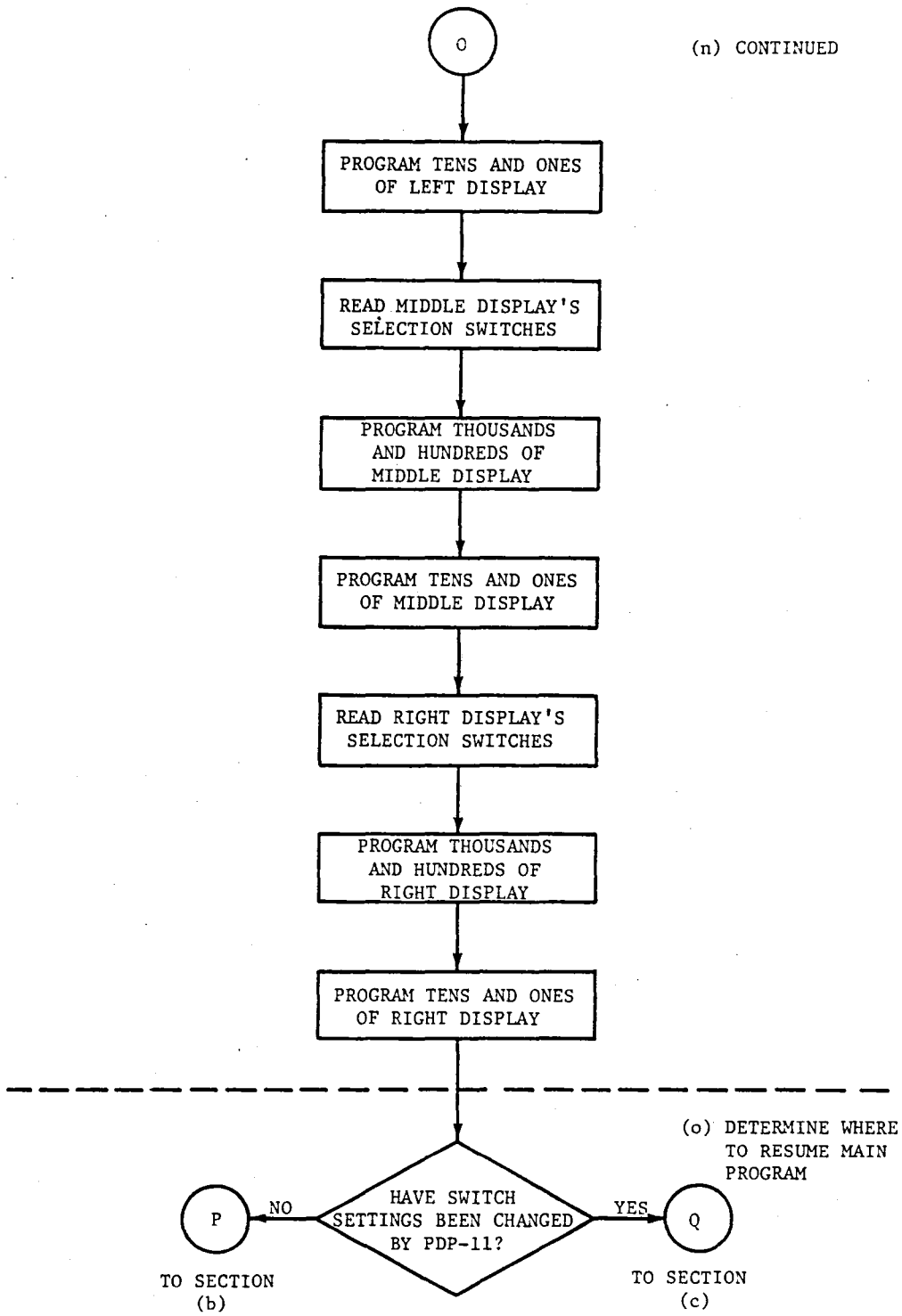


Figure C1. (concluded).

Table C1. Main program source listing.

```
;
;
;MAIN PROGRAM LISTING FOR LIDAR ENERGY MONITOR PROGRAM
;*****************************************************
;
;
P1ESUM  EQU     0800H
D1ESUM  EQU     0803H
X1ESUM  EQU     0806H
P2ESUM  EQU     0809H
D2ESUM  EQU     080CH
X2ESUM  EQU     080FH
P1ESM3  EQU     0802H
D1ESM3  EQU     0805H
X1ESM3  EQU     0808H
P2ESM3  EQU     080BH
D2ESM3  EQU     080EH
X2ESM3  EQU     0811H
NADDR   EQU     0815H
NUMFIR  EQU     0817H
P1RAVG  EQU     834H
D1RAVG  EQU     836H
X1RAVG  EQU     838H
P2RAVG  EQU     83AH
D2RAVG  EQU     83CH
X2RAVG  EQU     83EH
DIVIDE  EQU     0100H
REARRG  EQU     0140H
ADDREG  EQU     0120H
SWSEL1  EQU     0170H
SWSEL2  EQU     0280H
;
;INITIALIZE STACK AND SET INTERRUPT MODE  (a)
;
START:  EI                      ;ENABLE INTERRUPTS
        IM      1               ;INTERRUPT MODE 1
        LD      SP,951H   ;INITIALIZE STACK
;
;READ SWITCHES AND ARRANGE INTO NEW N  (b)
;
AVGRUN: IN      A,(0FH)    ;READ SWITCH SETTINGS
        RES     2,A        ;ARRANGE READINGS INTO
        RES     6,A        ;NEW N
        RES     7,A
        SRL     A
        SRL     A
        LD      (NADDR),A ;LOAD ADDRESS HOLDING
                           ;N WITH NEW N
```

(continued)

Table C1. (continued.)

```
;
;CALCULATE 2^N FOR # FIRINGS TO BE AVERAGED  (c)
;
CAL:    RES     4,A     ;RESET IN CASE SET
                        ;BY PDPSER
        LD      HL,00H  ;CLEAR HL
        LD      D,0     ;N=0?
        CP      D
        JP      Z,N0
        LD      D,1     ;N=1?
        CP      D
        JP      Z,N1
        LD      D,2     ;N=2?
        CP      D
        JP      Z,N2
        LD      D,3     ;N=3?
        CP      D
        JP      Z,N3
        LD      D,4     ;N=4?
        CP      D
        JP      Z,N4
        LD      D,5     ;N=5?
        CP      D
        JP      Z,N5
        LD      D,6     ;N=6?
        CP      D
        JP      Z,N6
        LD      D,7     ;N=7?
        CP      D
        JP      Z,N7
        LD      D,8     ;N=8?
        CP      D
        JP      Z,N8
        LD      D,9     ;N=9?
        CP      D
        JP      Z,N9
        LD      D,0AH   ;N=10?
        CP      D
        JP      Z,N10
        LD      D,0BH   ;N=11?
        CP      D
        JP      Z,N11
        LD      D,0CH   ;N=12?
        CP      D
        JP      Z,N12
        LD      D,0DH   ;N=13?
        CP      D
        JP      Z,N13
        SET     6,H     ;N=14, 2 TO N=16384
        JP      STORE
N0:     SET     0,L     ;N=0, 2 TO N=1
        JP      STORE
N1:     SET     1,L     ;N=1, 2 TO N=2
        JP      STORE
```

Table C1. (continued.)

```
N2:      SET     2,L       ;N=2,2 TO N=4
         JP      STORE
N3:      SET     3,L       ;N=3,2 TO N=8
         JP      STORE
N4:      SET     4,L       ;N=4,2 TO N=16
         JP      STORE
N5:      SET     5,L       ;N=5,2 TO N=32
         JP      STORE
N6:      SET     6,L       ;N=6,2 TO N=64
         JP      STORE
N7:      SET     7,L       ;N=7,2 TO N=128
         JP      STORE
N8:      SET     0,H       ;N=8,2 TO N=256
         JP      STORE
N9:      SET     1,H       ;N=9,2 TO N=512
         JP      STORE
N10:     SET     2,H       ;N=10,2 TO N=1024
         JP      STORE
N11:     SET     3,H       ;N=11,2 TO N=2048
         JP      STORE
N12:     SET     4,H       ;N=12,2 TO N=4096
         JP      STORE
N13:     SET     5,H       ;N=13,2 TO N=8192
STORE:   LD      (NUMFIR),HL
;
;CLEAR MEMORIES HOLDING CHANNEL ENERGY SUMS  (d)
;
INTRT1:  LD      C,11H       ;INITIALIZE COUNTER
         XOR     A           ;CLEAR ACCUMULATOR
         LD      DE,P1ESUM
CLRMEM:  LD      (DE),A
         CP      C
         JP      Z,J1
         DEC     C
         INC     DE
         JP      CLRMEM
;
;INITIALIZE SHOT COUNTER  (e)
;
J1:      LD      B,0
;
;CHECK IF LASER SYSTEM HAS FIRED  (f)
;
BSYWT:   IN      A,(06H)     ;READ A/D BUSY SIGNALS
         BIT     0,A
         JP      Z,J2        ;TEST P2 BUSY IF P1 BUSY
                             ;IS NOT HIGH
LOOP1:   IN      A,(06H)     ;LOOP TILL P1 BUSY GOES LOW
         BIT     0,A
         JP      NZ,LOOP1
         JP      J3
```

Table C1. (continued.)

```
J2:     BIT     3,A
        JP      Z,BSYWT     ;INPUT NEW BUSY SIGNALS IF
                            ;P2 HAS NOT FIRED
LOOP2:  IN      A,(06H)     ;LOOP TILL P2 BUSY GOES LOW
        BIT     3,A
        JP      NZ,LOOP2
;
;SUM CHANNEL ENERGIES TO PREVIOUS TOTALS  (≤)
;
J3:     PUSH    BC
        LD      C,0         ;P1 ENERGY SUMMING SEQUENCE
        IN      E,(C)
        LD      C,08H
        IN      D,(C)
        CALL    REARRG
        LD      HL,P1ESUM
        LD      C,E
        CALL    ADDREG
        LD      C,1         ;D1 ENERGY SUMMING SEQUENCE
        IN      E,(C)
        LD      C,09H
        IN      D,(C)
        CALL    REARRG
        LD      HL,D1ESUM
        LD      C,E
        CALL    ADDREG
        LD      C,2         ;X1 ENERGY SUMMING SEQUENCE
        IN      E,(C)
        LD      C,0AH
        IN      D,(C)
        CALL    REARRG
        LD      HL,X1ESUM
        LD      C,E
        CALL    ADDREG
        LD      C,3         ;P2 ENERGY SUMMING SEQUENCE
        IN      E,(C)
        LD      C,0BH
        IN      D,(C)
        CALL    REARRG
        LD      HL,P2ESUM
        LD      C,E
        CALL    ADDREG
        LD      C,4         ;D2 ENERGY SUMMING SEQUENCE
        IN      E,(C)
        LD      C,0CH
        IN      D,(C)
        CALL    REARRG
        LD      HL,D2ESUM
        LD      C,E
        CALL    ADDREG
        LD      C,5         ;X2 ENERGY SUMMING SEQUENCE
        IN      E,(C)
        LD      C,0DH
        IN      D,(C)
```

Table C1. (continued.)

```
        CALL    REARRG
        LD      HL,X2ESUM
        LD      C,E
        CALL    ADDREG
        POP     BC
;
;ADVANCE SHOT COUNTER  (h)
;
        INC     BC
;
;CALCULATE RUNNING AVERAGES FOR EACH CHANNEL  (i)
;
        LD      DE,(P1ESUM)    ;GET P1ESUM
        LD      HL,(P1ESM3)
        LD      H,0
        CALL    DIVIDE        ;CALCULATE P1 RUNNING AVG.
        LD      (P1RAVG),DE    ;STORE AT P1RAVG
        LD      DE,(D1ESUM)    ;GET D1ESUM
        LD      HL,(D1ESM3)
        LD      H,0
        CALL    DIVIDE        ;CALCULATE D1 RUNNING AVG.
        LD      (D1RAVG),DE     ;STORE AT D1RAVG
        LD      DE,(X1ESUM)     ;GET X1ESUM
        LD      HL,(X1ESM3)
        LD      H,0
        CALL    DIVIDE        ;CALCULATE X1 RUNNING AVG.
        LD      (X1RAVG),DE     ;STORE AT X1RAVG
        LD      DE,(P2ESUM)    ;GET P2ESUM
        LD      HL,(P2ESM3)
        LD      H,0
        CALL    DIVIDE        ;CALCULATE P2 RUNNING AVG.
        LD      (P2RAVG),DE    ;STORE AT P2RAVG
        LD      DE,(D2ESUM)    ;GET D2ESUM
        LD      HL,(D2ESM3)
        LD      H,0
        CALL    DIVIDE        ;CALCULATE D2 RUNNING AVG.
        LD      (D2RAVG),DE    ;STORE AT D2RAVG
        LD      DE,(X2ESUM)    ;GET X2ESUM
        LD      HL,(X2ESM3)
        LD      H,0
        CALL    DIVIDE        ;CALCULATE X2 RUNNING AVG.
        LD      (X2RAVG),DE    ;STORE AT X2RAVG
;
;CHECK IF N FIRINGS REACHED  (j)
;
        XOR     A             ;RESET CARRY FLAG
        LD      HL,(NUMFIR)
        SBC     HL,BC
        JP      NZ,BSYWT   ;WAIT FOR NEXT FIRING UNLESS BC=N
;
;SEND LOW BYTE OF P1 ENERGY AVG. TO PDP-11  (k)
;
```

Table C1. (continued.)

```
PDPRTN: LD      DE,P1RAVG
        LD      A,(DE)
        OUT     (00H),A
;
;WAIT 12 U-SEC FOR PDP-11 RESPONSE  (1)
;
        LD      C,2
LP1:    DEC     C
LP1A:   JP      Z,DISPLAY
LP1B:   JP      LP1
;
;TRANSFER ENERGY AVG. DATA TO PDP-11  (m)
;
RETRN:  INC     DE          ;HIGH BYTE OF P1 ENERGY
        LD      A,(DE)      ;AVG. TO PDP-11
        OUT     (01H),A
H1:     JR      H1
        LD      DE,D1RAVG ;LOW BYTE OF D1 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (00H),A
H2:     JR      H2
        INC     DE          ;HIGH BYTE OF D1 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (01H),A
H3:     JR      H3
        LD      DE,X1RAVG ;LOW BYTE OF X1 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (00H),A
H4:     JR      H4
        INC     DE          ;HIGH BYTE OF X1 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (01H),A
H5:     JR      H5
        LD      DE,P2RAVG ;LOW BYTE OF P2 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (00H),A
H6:     JR      H6
        INC     DE          ;HIGH BYTE OF P2 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (01H),A
H7:     JR      H7
        LD      DE,D2RAVG ;LOW BYTE OF D2 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (00H),A
H8:     JR      H8
        INC     DE          ;HIGH BYTE OF D2 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (01H),A
H9:     JR      H9
        LD      DE,X2RAVG ;LOW BYTE OF X2 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (00H),A
```

Table C1. (continued.)

```
H10:    JR      H10
        INC     DE          ;HIGH BYTE OF X2 ENERGY
        LD      A,(DE)      ;AVG. SENT TO PDP-11
        OUT     (01H),A
H11:    JR      H11
;
;PROGRAM 3-CHANNEL SELECTABLE DISPLAY  (n)
;
DISPLAY:IN      A,(07H)     ;READ LEFT DISPLAY'S
                            ;SELECTION SWITCHES
        RES     3,A
        RES     4,A
        RES     5,A
        RES     6,A
        RES     7,A
        CALL    SWSEL1
        LD      C,02H
        OUT     (C),D       ;PROGRAM THOUSANDS AND
                            ;HUNDREDS OF LEFT DISPLAY
        CALL    SWSEL2
        LD      C,03H
        OUT     (C),D       ;PROGRAM TENS AND ONES
                            ;OF LEFT DISPLAY
        IN      A,(07H)     ;READ MIDDLE DISPLAY'S
                            ;SELECTION SWITCHES
        RES     6,A
        RES     7,A
        SRL     A
        SRL     A
        SRL     A
        CALL    SWSEL1
        LD      C,04H
        OUT     (C),D       ;PROGRAM THOUSANDS AND
                            ;HUNDREDS OF MIDDLE DISPLAY
        CALL    SWSEL2
        LD      C,05H
        OUT     (C),D       ;PROGRAM TENS AND ONES
                            ;OF MIDDLE DISPLAY
        IN      A,(0FH)     ;READ RIGHT DISPLAY'S
                            ;SELECTION SWITCHES
        RES     3,A
        RES     4,A
        RES     5,A
        RES     6,A
        RES     7,A
        CALL    SWSEL1
        LD      C,06H
        OUT     (C),D       ;PROGRAM THOUSANDS AND
                            ;HUNDREDS OF RIGHT DISPLAY
        CALL    SWSEL2
        LD      C,07H
        OUT     (C),D       ;PROGRAM TENS AND ONES
                            ;OF RIGHT DISPLAY
;
;DETERMINE WHERE TO RESUME MAIN PROGRAM  (o)
;
```

Table C1. (concluded.)

```
LD      A,(NADDR)
BIT     4,A         ;CHECK IF SWITCHES HAVE BEEN
                    ;CHANGED BY PDP-11
JP      Z,AVGRUN    ;GO TO AVGRUN TO READ SWITCHES
                    ;IF SWITCHES UNCHANGED
JP      CAL         ;OTHERWISE, GO TO SEQUENCE
                    ;THAT DETERMINES NEW VALUE OF
                    ;2 RAISED TO N POWER
```

A.

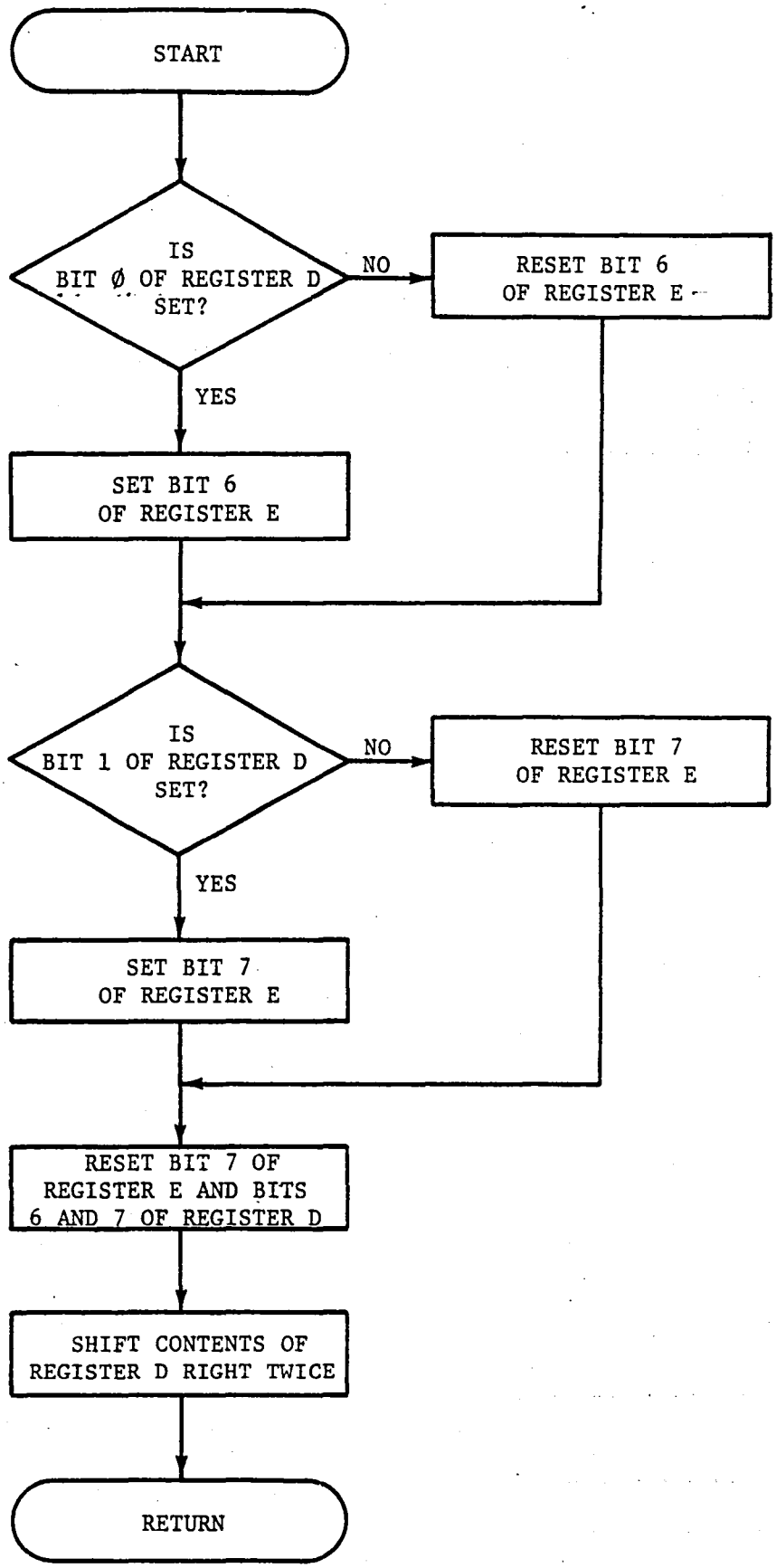


Figure C2. Flow chart for subroutine REARRG.

Table C2a. Subroutine source listing for REARRG.

```
;
;
;SUBROUTINE REARRG
;==================
;
;
;SUBROUTINE REARRG TAKES THE TWO BYTES
;CONTAINING THE ELEVEN BITS OF A GIVEN
;CHANNEL'S ENERGY FOR A FIRING,FROM THE
;A/D, AND REARRANGES THEM INTO THE
;CORRECT TWO BYTE BINARY NUMBER.
REARRG: BIT     0,D
        JP      Z,L1
        SET     6,E
        JP      L2
L1:     RES     6,E
L2:     BIT     1,D
        JP      Z,L3
        SET     7,E
        JP      L4
L3:     RES     7,E
L4:     RES     6,D
        RES     7,D
        SRL     D
        SRL     D
        RET

A.
```

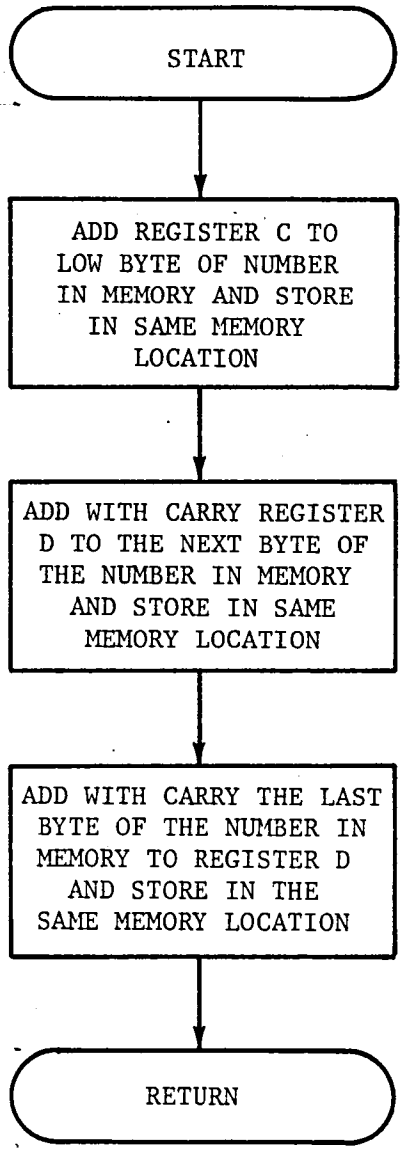


Figure C3. Flow chart for subroutine ADDREG.

Table C2b. Subroutine source listing for ADDREG.

```
;
;
;SUBROUTINE ADDREG
;==================
;
;THIS SUBROUTINE ADDS TWO TRIPLE PRECISION NUMBERS.
;BEFORE THIS ROUTINE IS CALLED, THE HL REGISTER
;PAIR MUST POINT TO THE LS-BYTE OF THE TRIPLE
;PRECISION VALUE STORED IN MEMORY.THE LS-BYTE
;OF THE OTHER TRIPLE PRECISION NUMBER MUST BE
;STORED IN REGISTER C AND THE NEXT SIGNIFICANT
;BYTE IN REGISTER D. THE MS-BYTE WILL ALWAYS BE
;00H AND IS TAKEN CARE OF WITHIN THE SUBROUTINE.
;
ADDREG: LD      A,C
        ADD     A,(HL)
        LD      (HL),A
        INC     HL
        LD      A,D
        ADC     A,(HL)
        LD      (HL),A
        INC     HL
        LD      A,00H
        ADC     A,(HL)
        LD      (HL),A
        RET
```

A.

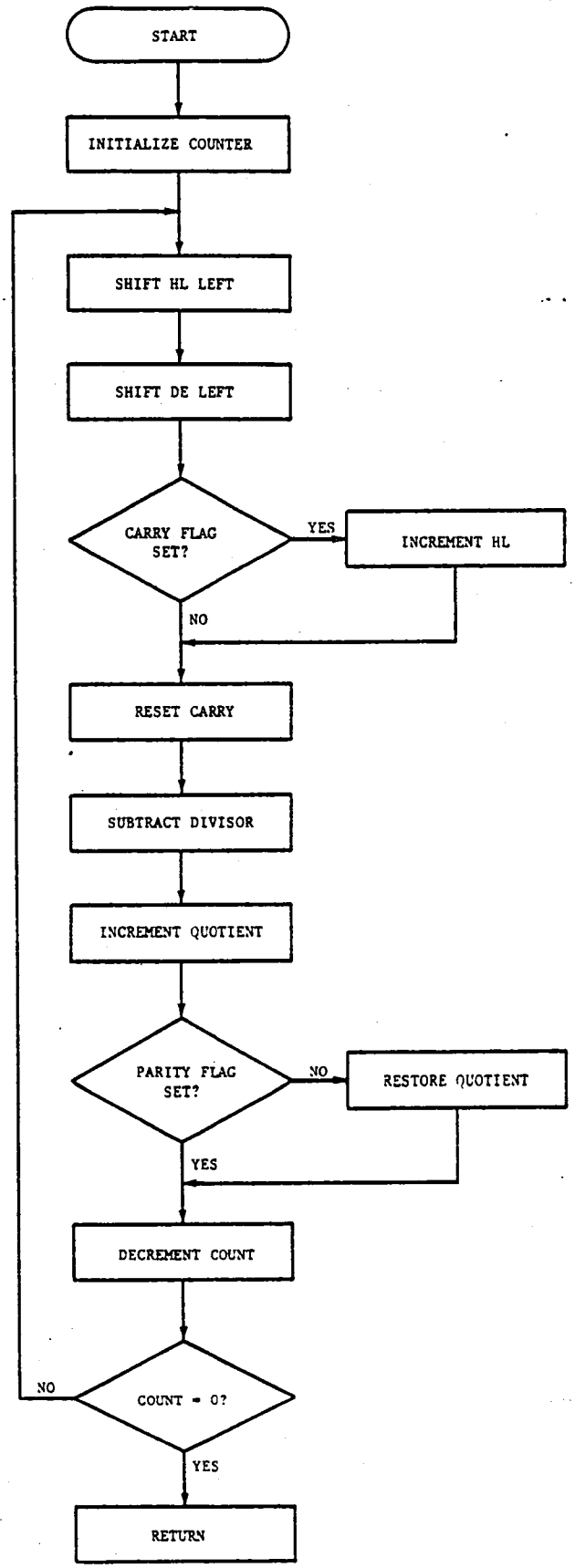


Figure C4. Flow chart for subroutine DIVIDE.

Table C2c. Subroutine source listing for DIVIDE.

```
;
;
;SUBROUTINE DIVIDE
;==================
;
;THIS SUBROUTINE PERFORMS A 32-BIT BY
;16-BIT UNSIGNED DIVIDE. THE 32-BIT
;DIVIDEND IS INPUT IN H,L,D,AND E
;(H=MSB,E=LSB), WHILE THE 16-BIT DIVISOR
;IS IN REG. PAIR BC. WHEN FINISHED THE
;16-BIT QUOTIENT IS HELD IN DE AND ANY
;REMAINDER IS IN HL.
;
DIVIDE: LD      A,16        ;ITERATION COUNTER
LOOP:   ADD     HL,HL       ;SHIFT HL LEFT
        EX      DE,HL       ;DE TO HL FOR SHIFT
        ADD     HL,HL       ;SHIFT (DE)
        EX      DE,HL
        JP      NC,JUMP1    ;GO IF NO CARRY
        INC     HL          ;CARRY TO MS 2 BYTES
JUMP1:  OR      A           ;RESET CARRY
        SBC     HL,BC       ;SUBTRACT DIVISOR
        INC     DE          ;SET Q=1
        JP      P,JUMP2     ;GO IF Q=1
        ADD     HL,BC       ;RESTORE
        RES     0,E         ;SET Q=0
JUMP2:  DEC     A           ;DECREMENT COUNT
        JP      NZ,LOOP     ;GO IF NOT DONE
        RET                 ;RETURN

A.
```

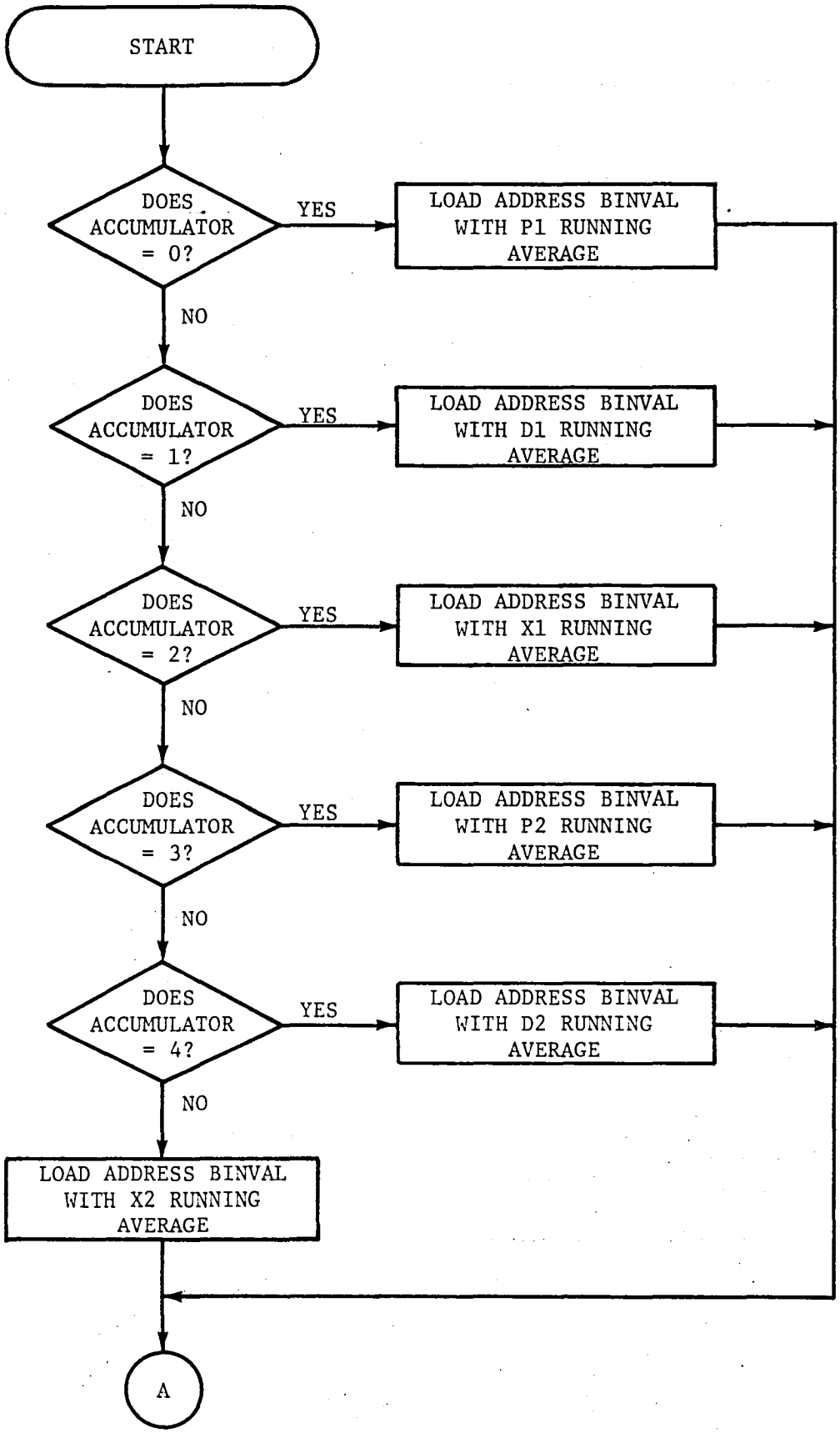


Figure C5. Flow chart for subroutine SWSEL1.

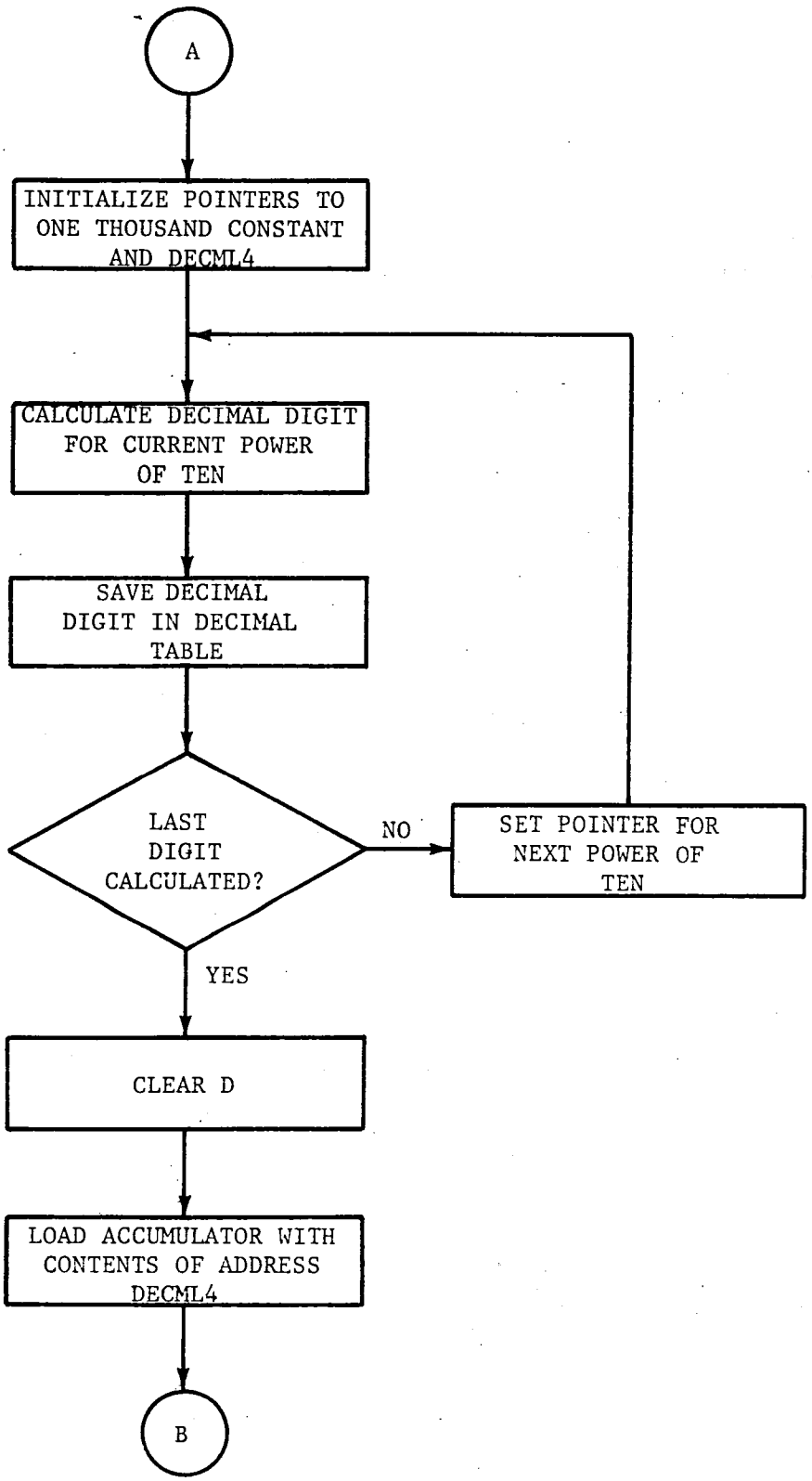


Figure C5. (continued).

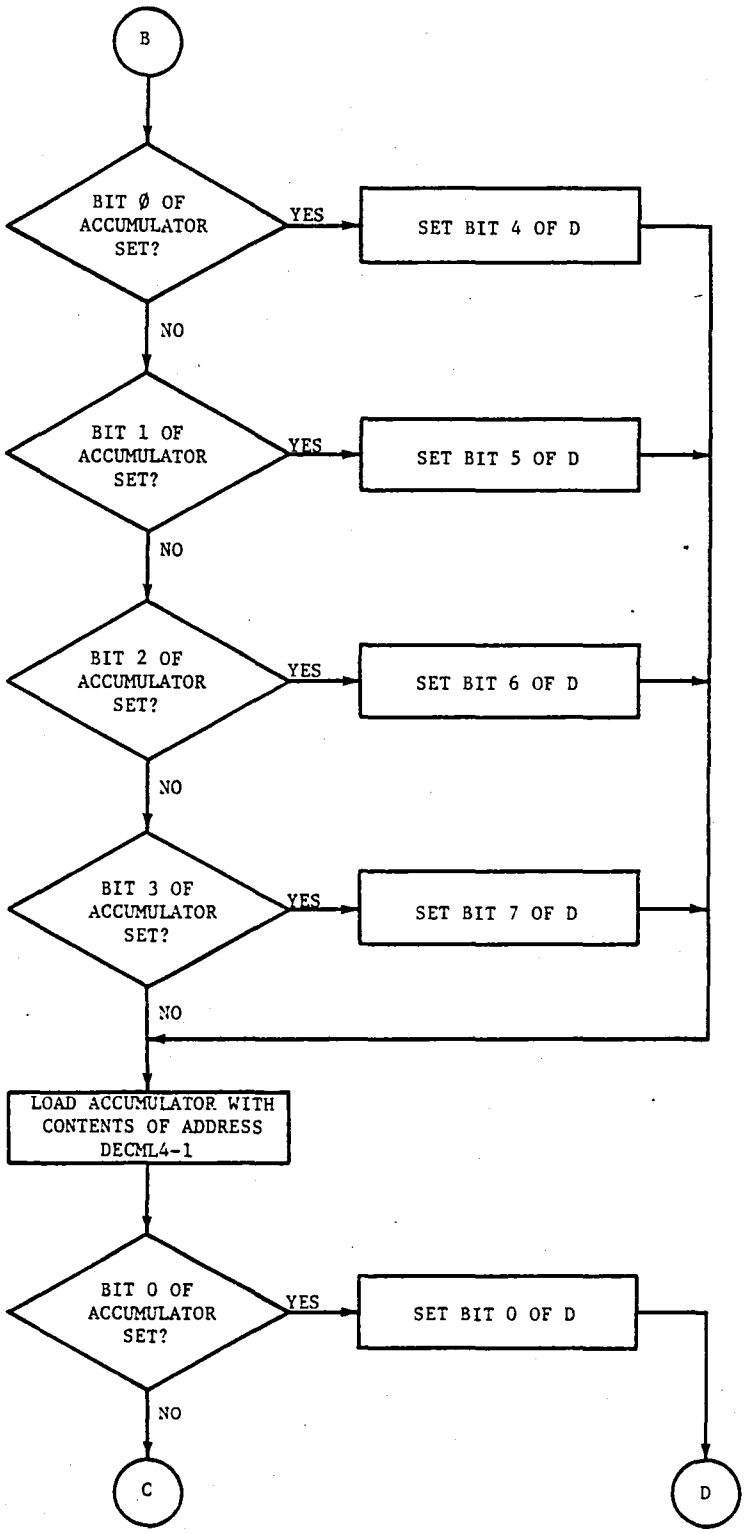


Figure C5. (continued).

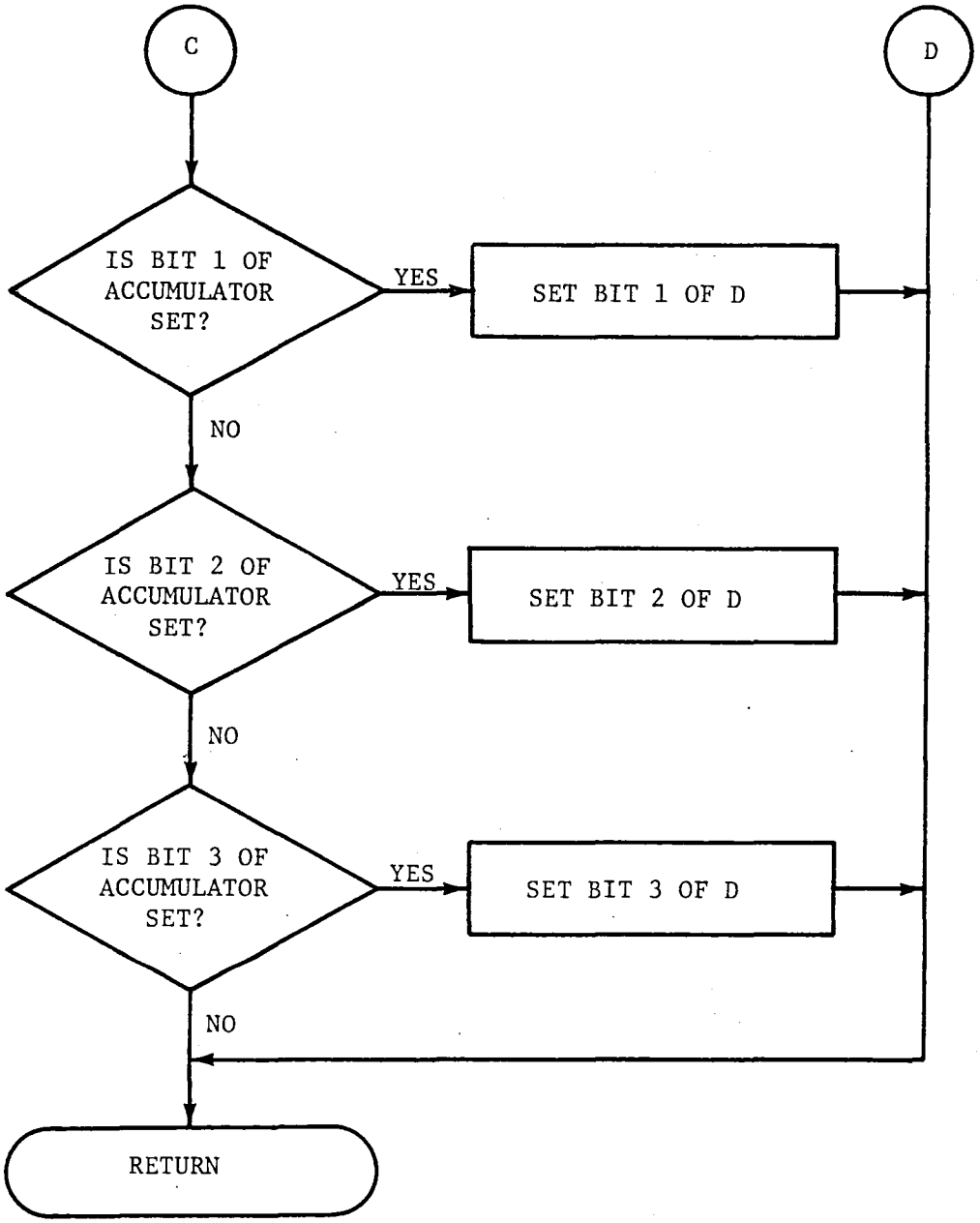


Figure C5. (concluded).

Table C2d. Subroutine source listing for SWSEL1.

```
;
;
;SUBROUTINE SWSEL1
;==================
;
;SUBROUTINE SWSEL1 DETERMINES WHICH CHANNEL
;TO DISPLAY ON THE DISPLAY BEING CONSIDERED,
;CONVERTS THE ENERGY AVERAGE TO BCD, THEN
;PROGRAMS THE THOUSANDS AND HUNDREDS
;DIGITS OF THE DISPLAY
;
THOUSN  EQU     0349H
BINVAL  EQU     0827H
P1RAVG  EQU     834H
D1RAVG  EQU     836H
X1RAVG  EQU     838H
P2RAVG  EQU     83AH
D2RAVG  EQU     83CH
X2RAVG  EQU     83EH
DECML4  EQU     0823H
SWSEL1: LD      D,00H       ;DISPLAY P1 ENERGY AVG. ?
        CP      D
        JP      Z,S1
        LD      D,01H       ;DISPLAY D1 AVG. ?
        CP      D
        JP      Z,S2
        LD      D,02H       ;DISPLAY X1 AVG. ?
        CP      D
        JP      Z,S3
        LD      D,03H       ;DISPLAY P2 AVG. ?
        CP      D
        JP      Z,S4
        LD      D,04H       ;DISPLAY D2 AVG. ?
        CP      D
        JP      Z,S5
        LD      DE,(X2RAVG)    ;DISPLAY X2 AVG.
        LD      (BINVAL),DE
        JP      BNTODC
S1:     LD      DE,(P1RAVG)    ;DISPLAY P1 AVG.
        LD      (BINVAL),DE
        JP      BNTODC
S2:     LD      DE,(D1RAVG)    ;DISPLAY D1 AVG.
        LD      (BINVAL),DE
        JP      BNTODC
S3:     LD      DE,(X1RAVG)    ;DISPLAY X1 AVG.
        LD      (BINVAL),DE
        JP      BNTODC
S4:     LD      DE,(P2RAVG)    ;DISPLAY P2 AVG.
        LD      (BINVAL),DE
        JP      BNTODC
```

Table C2d. (continued.)

```
S5:      LD      DE,(D2RAVG)    ;DISPLAY D2 AVG.
         LD      (BINVAL),DE
BNTODC:  LD      HL,THOUSN      ;THE ROUTINE BNTODC IS
                                ;DESCRIBED IN THE
                                ;Z80 COOKBOOK, PG. 178
         LD      DE,DECML4
BNDC:    PUSH    HL
         PUSH    DE
DCEQVL:  LD      DE,BINVAL
         LD      BC,0200H
DCLOOP:  AND     A
DCLP1:   LD      A,(DE)
         SBC     A,(HL)
         LD      (DE),A
         DEC     B
         JP      Z,INCRVL
         INC     HL
         INC     DE
         JP      DCLP1
INCRVL:  INC     C
         DEC     HL
         LD      DE,BINVAL
         LD      B,02H
         JP      NC,DCLOOP
         DEC     C
         EX      DE,HL
         PUSH    BC
         LD      BC,0002H
ADDER:   AND     A
ADDMOR:  LD      A,(DE)
         ADC     A,(HL)
         LD      (HL),A
         CPI
         JP      PO,NEXT
         INC     DE
         JP      ADDMOR
NEXT:    POP     BC
         POP     HL
         LD      (HL),C
         EX      DE,HL
         POP     HL
         INC     HL
         INC     HL
         LD      A,E
         CP      20H
         JP      Z,A0
         DEC     DE
         JP      BNDC        ;END OF BNTODC
A0:      LD      D,0         ;ARRANGE BITS
         LD      A,(DECML4)  ;4-7 OF REG. D
                             ;TO MATCH THE
         BIT     0,A         ;BCD NUMBER AT
         JP      Z,A1        ;ADDRESS DECML4
```

Table C2d. (concluded.)

```
        SET     4,D
A1:     BIT     1,A
        JP      Z,A2
        SET     5,D
A2:     BIT     2,A
        JP      Z,A3
        SET     6,D
A3:     BIT     3,A
        JP      Z,A4
        SET     7,D
A4:     LD      HL,DECML4  ;ARRANGE BITS
        DEC     HL         ;0-3 OF REG. D
        LD      A,(HL)     ;TO MATCH THE
        BIT     0,A        ;BCD NUMBER AT
        JP      Z,A5       ;ADDRESS DECML4-1
        SET     0,D
A5:     BIT     1,A
        JP      Z,A6
        SET     1,D
A6:     BIT     2,A
        JP      Z,A7
        SET     2,D
A7:     BIT     3,A
        JP      Z,A8
        SET     3,D
A8:     RET

A.
```

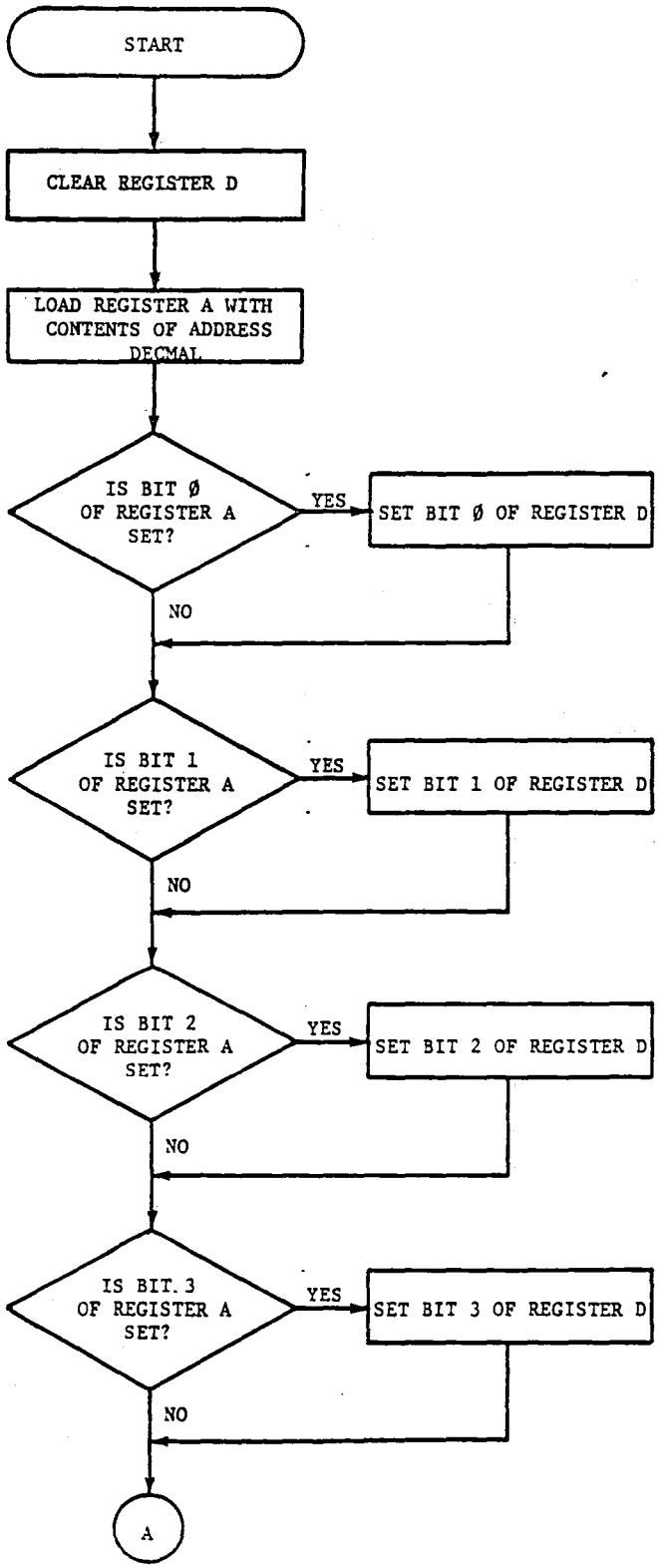


Figure C6. Flow chart for subroutine SWSEL2.

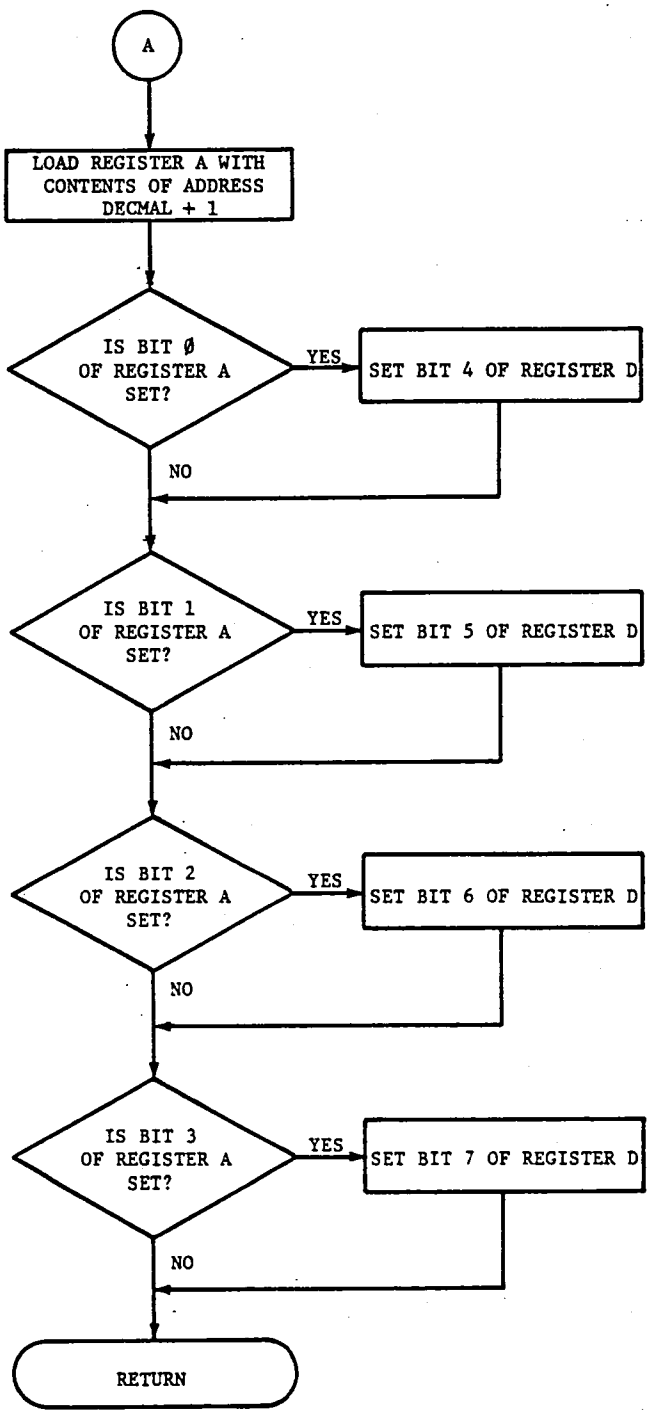


Figure C6. (concluded.)

Table C2e. Subroutine source listing for SWSEL2.

```
;
;
;SUBROUTINE SWSEL2
;==================
;
;SUBROUTINE SWSEL2 PROGRAMS THE TENS
;AND ONES DIGITS OF THE DISPLAY UNDER
;CONSIDERATION WHEN IT IS CALLED WITH
;THE DATA FOR THE CHANNEL SELECTED IN
;SUBROUTINE SWSEL1.
;
DECMAL  EQU     820H
;
SWSEL2: LD      D,0         ;ARRANGE BITS
        LD      A,(DECMAL)  ;0-3 OF REG. D
                            ;TO MATCH THE
        BIT     0,A         ;BCD NUMBER AT
        JP      Z,B1        ;ADDRESS DECMAL
        SET     0,D         ;WHICH HOLDS THE
B1:     BIT     1,A         ;ONES DIGIT
        JP      Z,B2
        SET     1,D
B2:     BIT     2,A
        JP      Z,B3
        SET     2,D
B3:     BIT     3,A
        JP      Z,B4
        SET     3,D
B4:     LD      HL,DECMAL   ;ARRANGE BITS
        INC     HL          ;4-7 OF REG. D
        LD      A,(HL)      ;TO MATCH THE
        BIT     0,A         ;BCD NUMBER AT
        JP      Z,B5        ;ADDRESS DECMAL+1
        SET     4,D         ;WHICH HOLDS THE
B5:     BIT     1,A         ;TENS DIGIT
        JP      Z,B6
        SET     5,D
B6:     BIT     2,A
        JP      Z,B7
        SET     6,D
B7:     BIT     3,A
        JP      Z,B8
        SET     7,D
B8:     RET

A.
```

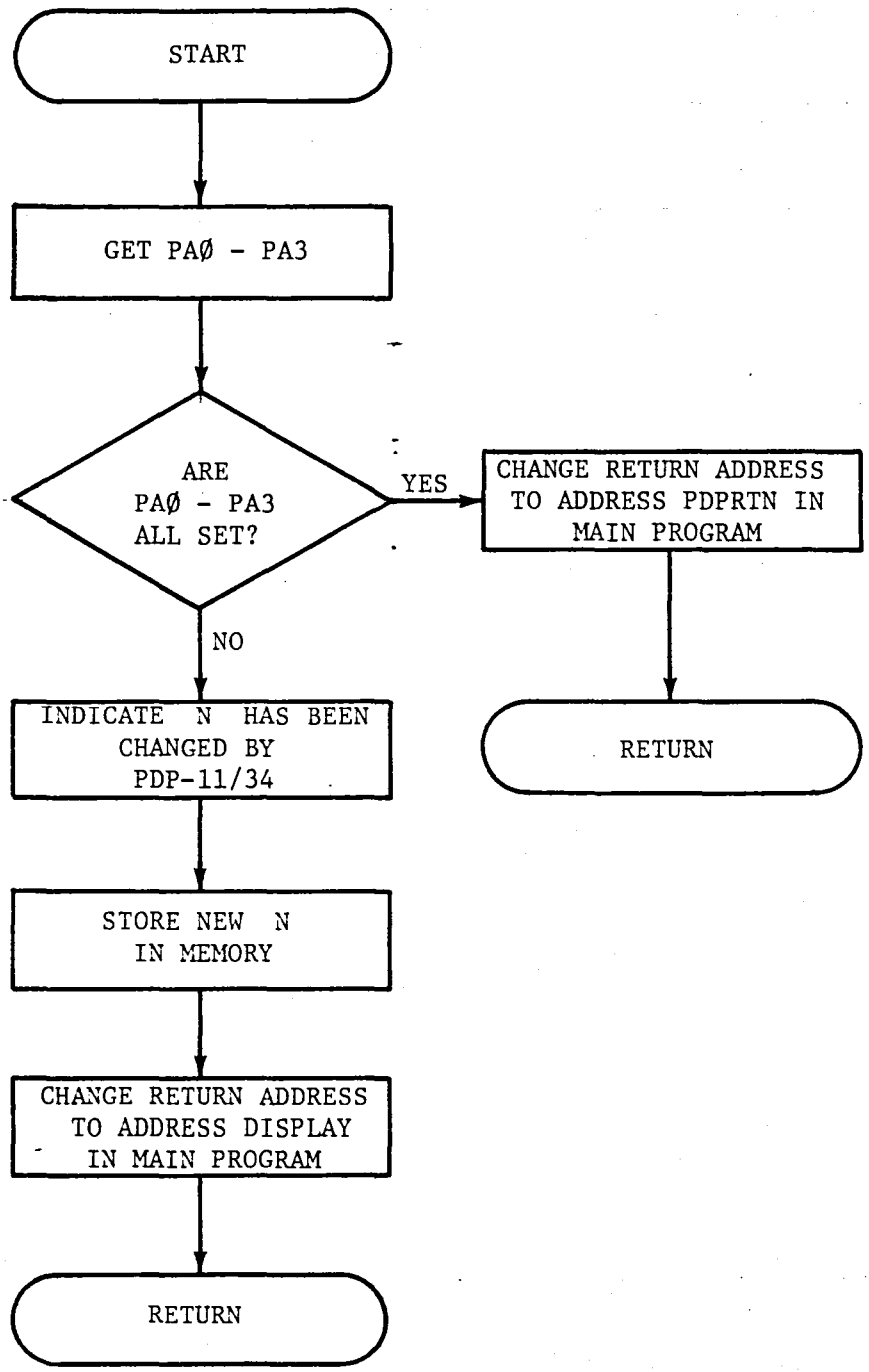


Figure C7. Flow chart for service routine PDPSER.

Table C2f. Subroutine source listing for PDPSER.

```
;
;INTERRUPT SERVICE ROUTINE PDPSER
;-----------------------------------
;
;UPON AN NMI INTERRUPT BY THE PDP-11, THE Z80 INPUTS DATA
;BITS PA0-PA3 FROM THE PDP-11 AND ARRANGES THEM AS THE
;LOWEST ORDER BITS OF AN 8-BIT WORD; THE HIGH ORDER BITS
;ARE ALL RESET. IF BITS PA0-PA3 ARE ALL SET , THEN THE
;PDP-11 HAS NOT CHANGED THE THE VALUE OF N BUT REQUIRES
;THE CURRENT RUNNING AVERAGE FOR EACH CHANNEL. OTHERWISE,
;BITS PA0-PA3 REPRESENT N AND BIT 4 IS SET TO INDICATE
;THAT THE PDP-11 PROVIDES N AS OPPOSED TO THE SWITCHES.
;
NADDR   EQU     815H
DISPLAY EQU     61BH
PDPRTN  EQU     5C0H
;
PDPSER: IN      A,(0EH)     ;GET PA0-PA3 FROM PDP-11
        RES     4,A         ;CLEAR HIGH ORDER BITS
        RES     5,A
        RES     6,A
        RES     7,A
        CP      0FH
        JP      Z,NOCHNG    ;LEAVE SERVICE ROUTINE TO OUTPUT
                            ;RUNNING AVERAGES IF N NOT
                            ;CHANGED BY PDP-11
        SET     4,A         ;INDICATE N CHANGED BY PDP-11
        LD      (NADDR),A   ;STORE NEW N
        LD      HL,DISPLAY  ;RETURN TO ADDRESS DISPLAY
        EX      (SP),HL     ;IN MAIN PROGRAM
        RETN
NOCHNG: LD      HL,PDPRTN   ;RETURN TO ADDRESS PDPRTN
        EX      (SP),HL     ;IN MAIN PROGRAM
        RETN

A.
```

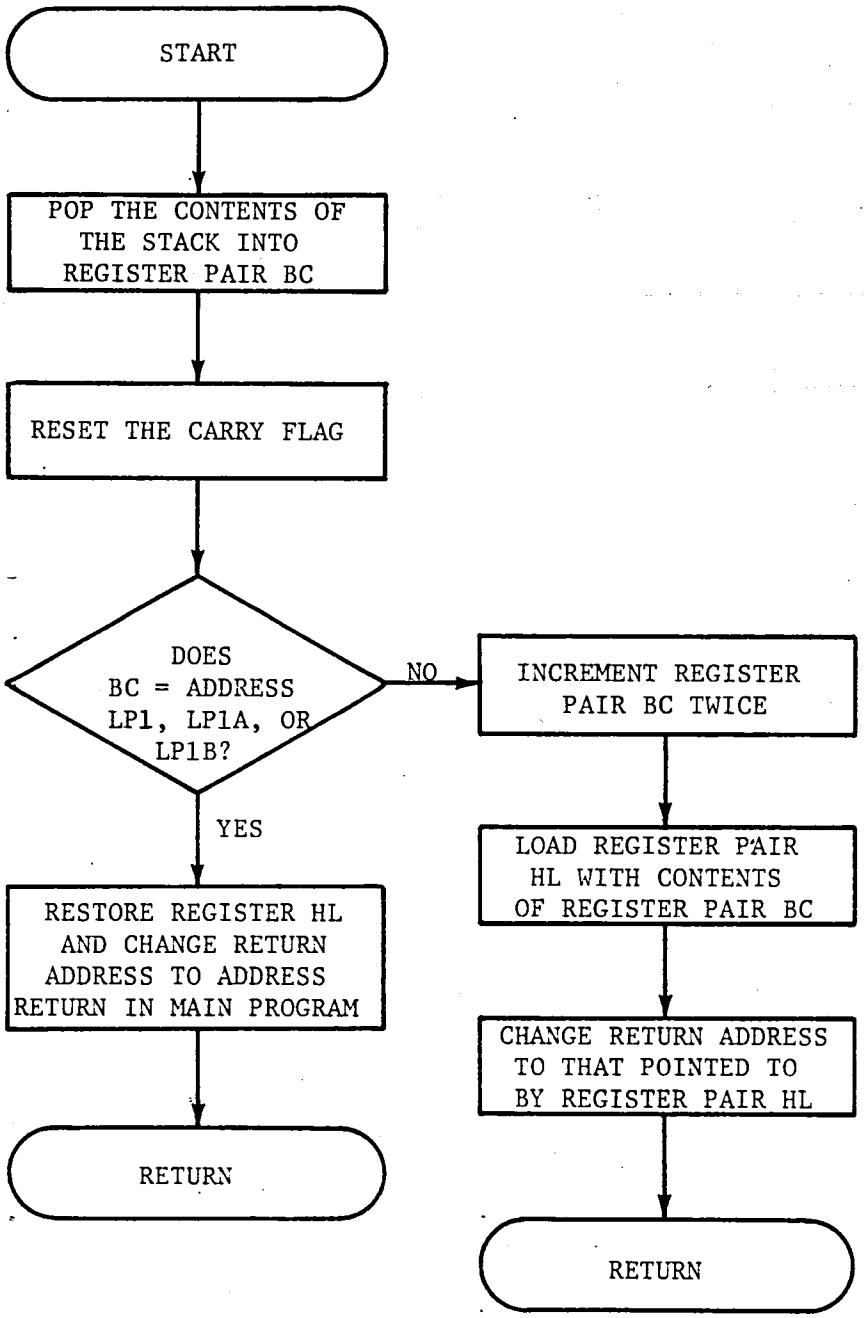


Figure C8. Flow chart for service routine DATRNSFR.

Table C2g. Subroutine source listing for DATRNSFR.

```
;
;
;INTERRUPT SERVICE ROUTINE DATRNSFR
;-----------------------------------
;
;THIS INTERRUPT SERVICE ROUTINE HANDLES THE TRANSFER OF DATA
;FROM THE Z80 TO THE PDP-11 V1A HANDSHAKING.
;
LP1     EQU     5C8H
LP1A    EQU     5C9H
LP1B    EQU     5CCH
RETRN   EQU     5CFH
;
TRNSFR: POP     BC          ;GET CURRENT RETURN ADDRESS
        XOR     A           ;RESET CARRY FLAG
        LD      HL,LP1
        SBC     HL,BC       ;USE CURRENT RETURN ADDRESS TO
        JP      Z,RET2      ;DETERMINE NEW RETURN ADDRESS
        XOR     A
        LD      HL,LP1A
        SBC     HL,BC
        JP      Z,RET2
        XOR     A
        LD      HL,LP1B
        SBC     HL,BC
        JP      Z,RET2
RET1:   INC     BC          ;CHANGE RETURN ADDRESS
        INC     BC
        LD      H,B
        LD      L,C
        DEC     SP
        DEC     SP
        EX      (SP),HL
        EI
        RETI
RET2:   LD      HL,RETRN ;CHANGE RETURN ADDRESS
        DEC     SP
        DEC     SP
        EX      (SP),HL
        EI
        RETI

A,
```

APPENDIX D

PRINT LISTINGS OF ASSEMBLED PROGRAMS

This Page Intentionally Left Blank

Table D1. Assembled listing of main program.

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;MAIN PROGRAM LISTING FOR LIDAR ENERGY MONITOR PROGRAM
                    0004 ;*****************************************************
                    0005 ;
                    0006 ;
      (0800)        0007 P1ESUM  EQU     0800H
      (0803)        0008 D1ESUM  EQU     0803H
      (0806)        0009 X1ESUM  EQU     0806H
      (0809)        0010 P2ESUM  EQU     0809H
      (080C)        0011 D2ESUM  EQU     080CH
      (080F)        0012 X2ESUM  EQU     080FH
      (0802)        0013 P1ESM3  EQU     0802H
      (0805)        0014 D1ESM3  EQU     0805H
      (0808)        0015 X1ESM3  EQU     0808H
      (080B)        0016 P2ESM3  EQU     080BH
      (080E)        0017 D2ESM3  EQU     080EH
      (0811)        0018 X2ESM3  EQU     0811H
      (0815)        0019 NADDR   EQU     0815H
      (0817)        0020 NUMFIR  EQU     0817H
      (0834)        0021 P1RAVG  EQU     834H
      (0836)        0022 D1RAVG  EQU     836H
      (0838)        0023 X1RAVG  EQU     838H
      (083A)        0024 P2RAVG  EQU     83AH
      (083C)        0025 D2RAVG  EQU     83CH
      (083E)        0026 X2RAVG  EQU     83EH
      (0100)        0027 DIVIDE  EQU     0100H
      (C140)        0028 REARRG  EQU     0140H
      (0120)        0029 ADDREG  EQU     0120H
      (0170)        0030 SWSEL1  EQU     0170H
      (0280)        0031 SWSEL2  EQU     0280H
                    0032 ;
                    0033 ;INITIALIZE STACK AND SET INTERRUPT MODE  (a)
                    0034 ;
0400  FB            0035 START:  EI                 ;ENABLE INTERRUPTS
0401  ED56          0036         IM      1          ;INTERRUPT MODE 1
0403  315109        0037         LD      SP,951H  ;INITIALIZE STACK
                    0038 ;
                    0039 ;READ SWITCHES AND ARRANGE INTO NEW N  (b)
                    0040 ;
0406  DB0F          0041 AVGRUN: IN      A,(0FH)    ;READ SWITCH SETTINGS
0408  CB97          0042         RES     2,A        ;ARRANGE READINGS INTO
040A  CBB7          0043         RES     6,A        ;NEW N
040C  CBBF          0044         RES     7,A
040E  CB3F          0045         SRL     A
0410  CB3F          0046         SRL     A
0412  321508        0047         LD      (NADDR),A ;LOAD ADDRESS HOLDING
                    0048                           ;N WITH NEW N
                    0049 ;            N
                    0050 ;CALCULATE 2  FOR # FIRINGS TO BE AVERAGED  (c)
                    0051 ;
0415  CBA7          0052 CAL:    RES     4,A      ;RESET IN CASE SET
                    0053                          ;BY PDPSER
0417  210000        0054         LD      HL,00H   ;CLEAR HL
041A  1600          0055         LD      D,0      ;N=0?
041C  BA            0056         CP      D
041D  CA7304        0057         JP      Z,NO
```

Table D1. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0002

0420  1601      0058          LD      D,1     ;N=1?
0422  BA        0059          CP      D
0423  CA7804    0060          JP      Z,N1
0426  1602      0061          LD      D,2     ;N=2?
0428  BA        0062          CP      D
0429  CA7D04    0063          JP      Z,N2
042C  1603      0064          LD      D,3     ;N=3?
042E  BA        0065          CP      D
042F  CA8204    0066          JP      Z,N3
0432  1604      0067          LD      D,4     ;N=4?
0434  BA        0068          CP      D
0435  CA8704    0069          JP      Z,N4
0438  1605      0070          LD      D,5     ;N=5?
043A  BA        0071          CP      D
043B  CA8C04    0072          JP      Z,N5
043E  1606      0073          LD      D,6     ;N=6?
0440  BA        0074          CP      D
0441  CA9104    0075          JP      Z,N6
0444  1607      0076          LD      D,7     ;N=7?
0446  BA        0077          CP      D
0447  CA9604    0078          JP      Z,N7
044A  1608      0079          LD      D,8     ;N=8?
044C  BA        0080          CP      D
044D  CA9B04    0081          JP      Z,N8
0450  1609      0082          LD      D,9     ;N=9?
0452  BA        0083          CP      D
0453  CAA004    0084          JP      Z,N9
0456  160A      0085          LD      D,0AH   ;N=10?
0458  BA        0086          CP      D
0459  CAA504    0087          JP      Z,N10
045C  160B      0088          LD      D,0BH   ;N=11?
045E  BA        0089          CP      D
045F  CAAA04    0090          JP      Z,N11
0462  160C      0091          LD      D,0CH   ;N=12?
0464  BA        0092          CP      D
0465  CAAF04    0093          JP      Z,N12
0468  160D      0094          LD      D,0DH   ;N=13?
046A  BA        0095          CP      D
046B  CAB404    0096          JP      Z,N13
046E  CBF4      0097          SET     6,H     ;N=14, 2 TO N=16384
0470  C3B604    0098          JP      STORE
0473  CBC5      0099 N0:      SET     0,L     ;N=0, 2 TO N=1
0475  C3B604    0100          JP      STORE
0478  CBCD      0101 N1:      SET     1,L     ;N=1, 2 TO N=2
047A  C3B604    0102          JP      STORE
047D  CBD5      0103 N2:      SET     2,L     ;N=2,2 TO N=4
047F  C3B604    0104          JP      STORE
0482  CBDD      0105 N3:      SET     3,L     ;N=3,2 TO N=8
0484  C3B604    0106          JP      STORE
0487  CBE5      0107 N4:      SET     4,L     ;N=4,2 TO N=16
0489  C3B604    0108          JP      STORE
048C  CBED      0109 N5:      SET     5,L     ;N=5,2 TO N=32
048E  C3B604    0110          JP      STORE
0491  CBF5      0111 N6:      SET     6,L     ;N=6,2 TO N=64
0493  C3B604    0112          JP      STORE
0496  CBFD      0113 N7:      SET     7,L     ;N=7,2 TO N=128
0498  C3B604    0114          JP      STORE
```

Table D1. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0003

049B  CBC4        0115 N8:     SET     0,H       ;N=8,2 TO N=256
049D  C3B604      0116         JP      STORE
04A0  CBCC        0117 N9:     SET     1,H       ;N=9,2 TO N=512
04A2  C3B604      0118         JP      STORE
04A5  CBD4        0119 N10:    SET     2,H       ;N=10,2 TO N=1024
04A7  C3B604      0120         JP      STORE
04AA  CBDC        0121 N11:    SET     3,H       ;N=11,2 TO N=2048
04AC  C3B604      0122         JP      STORE
04AF  CBE4        0123 N12:    SET     4,H       ;N=12,2 TO N=4096
04B1  C3B604      0124         JP      STORE
04B4  CBEC        0125 N13:    SET     5,H       ;N=13,2 TO N=8192
04B6  221708      0126 STORE:  LD      (NUMFIR),HL
                  0127 ;
                  0128 ;CLEAR MEMORIES HOLDING CHANNEL ENERGY SUMS  (d)
                  0129 ;
04B9  0E11        0130 INTRT1: LD      C,11H       ;INITIALIZE COUNTER
04BB  AF          0131         XOR     A           ;CLEAR ACCUMULATOR
04BC  110008      0132         LD      DE,P1ESUM
04BF  12          0133 CLRMEM: LD      (DE),A
04C0  B9          0134         CP      C
04C1  CAC904      0135         JP      Z,J1
04C4  0D          0136         DEC     C
04C5  13          0137         INC     DE
04C6  C3BF04      0138         JP      CLRMEM
                  0139 ;
                  0140 ;INITIALIZE SHOT COUNTER  (e)
                  0141 ;
04C9  0600        0142 J1:     LD      B,0
                  0143 ;
                  0144 ;CHECK IF LASER SYSTEM HAS FIRED  (f)
                  0145 ;
04CB  DB06        0146 BSYWT:  IN      A,(06H)     ;READ A/D BUSY SIGNALS
04CD  CB47        0147         BIT     0,A
04CF  CADC04      0148         JP      Z,J2        ;TEST P2 BUSY IF P1 BUSY
                  0149                             ;IS NOT HIGH
04D2  DB06        0150 LOOP1:  IN      A,(06H)     ;LOOP TILL P1 BUSY GOES LOW
04D4  CB47        0151         BIT     0,A
04D6  C2D204      0152         JP      NZ,LOOP1
04D9  C3E804      0153         JP      J3
04DC  CB5F        0154 J2:     BIT     3,A
04DE  CACB04      0155         JP      Z,BSYWT     ;INPUT NEW BUSY SIGNALS IF
                  0156                             ;P2 HAS NOT FIRED
04E1  DB06        0157 LOOP2:  IN      A,(06H)     ;LOOP TILL P2 BUSY GOES LOW
04E3  CB5F        0158         BIT     3,A
04E5  C2E104      0159         JP      NZ,LOOP2
                  0160 ;
                  0161 ;SUM CHANNEL ENERGIES TO PREVIOUS TOTALS  (g)
                  0162 ;
04E8  C5          0163 J3:     PUSH    BC
04E9  0E00        0164         LD      C,0         ;P1 ENERGY SUMMING SEQUENCE
04EB  ED58        0165         IN      E,(C)
04ED  0E08        0166         LD      C,08H
04EF  ED50        0167         IN      D,(C)
04F1  CD4001      0168         CALL    REARRG
04F4  210008      0169         LD      HL,P1ESUM
04F7  4B          0170         LD      C,E
04F8  CD2001      0171         CALL    ADDREG
```

Table D1. (continued.)


```
04FB  0E01        0172        LD      C,1         ;D1 ENERGY SUMMING SEQUENCE
04FD  ED58        0173        IN      E,(C)
04FF  0E09        0174        LD      C,09H
0501  ED50        0175        IN      D,(C)
0503  CD4001      0176        CALL    REARRG
0506  210308      0177        LD      HL,D1ESUM
0509  4B          0178        LD      C,E
050A  CD2001      0179        CALL    ADDREG
050D  0E02        0180        LD      C,2         ;X1 ENERGY SUMMING SEQUENCE
050F  ED58        0181        IN      E,(C)
0511  0E0A        0182        LD      C,0AH
0513  ED50        0183        IN      D,(C)
0515  CD4001      0184        CALL    REARRG
0518  210608      0185        LD      HL,X1ESUM
051B  4B          0186        LD      C,E
051C  CD2001      0187        CALL    ADDREG
051F  0E03        0188        LD      C,3         ;P2 ENERGY SUMMING SEQUENCE
0521  ED58        0189        IN      E,(C)
0523  0E0B        0190        LD      C,0BH
0525  ED50        0191        IN      D,(C)
0527  CD4001      0192        CALL    REARRG
052A  210908      0193        LD      HL,P2ESUM
052D  4B          0194        LD      C,E
052E  CD2001      0195        CALL    ADDREG
0531  0E04        0196        LD      C,4         ;D2 ENERGY SUMMING SEQUENCE
0533  ED58        0197        IN      E,(C)
0535  0E0C        0198        LD      C,0CH
0537  ED50        0199        IN      D,(C)
0539  CD4001      0200        CALL    REARRG
053C  210C08      0201        LD      HL,D2ESUM
053F  4B          0202        LD      C,E
0540  CD2001      0203        CALL    ADDREG
0543  0E05        0204        LD      C,5         ;X2 ENERGY SUMMING SEQUENCE
0545  ED58        0205        IN      E,(C)
0547  0E0D        0206        LD      C,0DH
0549  ED50        0207        IN      D,(C)
054B  CD4001      0208        CALL    REARRG
054E  210F08      0209        LD      HL,X2ESUM
0551  4B          0210        LD      C,E
0552  CD2001      0211        CALL    ADDREG
0555  C1          0212        POP     BC
                  0213 ;
                  0214 ;ADVANCE SHOT COUNTER  (h)
                  0215 ;
0556  03          0216        INC     BC
                  0217 ;
                  0218 ;CALCULATE RUNNING AVERAGES FOR EACH CHANNEL  (i)
                  0219 ;
0557  ED5B0008    0220        LD      DE,(P1ESUM)    ;GET P1ESUM
055B  2A0208      0221        LD      HL,(P1ESM3)
055E  2600        0222        LD      H,0
0560  CD0001      0223        CALL    DIVIDE       ;CALCULATE P1 RUNNING AVG.
0563  ED533408    0224        LD      (P1RAVG),DE    ;STORE AT P1RAVG
0567  ED5B0308    0225        LD      DE,(D1ESUM)    ;GET D1ESUM
056B  2A0508      0226        LD      HL,(D1ESM3)
056E  2600        0227        LD      H,0
0570  CD0001      0228        CALL    DIVIDE       ;CALCULATE D1 RUNNING AVG.
```

Table D1. (continued.)


```
0573  ED533608     0229          LD      (D1RAVG),DE     ;STORE AT D1RAVG
0577  ED5B0608     0230          LD      DE,(X1ESUM)     ;GET X1ESUM
057B  2A0808       0231          LD      HL,(X1ESM3)
057E  2600         0232          LD      H,0
0580  CD0001       0233          CALL    DIVIDE        ;CALCULATE X1 RUNNING AVG.
0583  ED533808     0234          LD      (X1RAVG),DE     ;STORE AT X1RAVG
0587  ED5B0908     0235          LD      DE,(P2ESUM)    ;GET P2ESUM
058B  2A0B08       0236          LD      HL,(P2ESM3)
058E  2600         0237          LD      H,0
0590  CD0001       0238          CALL    DIVIDE        ;CALCULATE P2 RUNNING AVG.
0593  ED533A08     0239          LD      (P2RAVG),DE    ;STORE AT P2RAVG
0597  ED5B0C08     0240          LD      DE,(D2ESUM)    ;GET D2ESUM
059B  2A0E08       0241          LD      HL,(D2ESM3)
059E  2600         0242          LD      H,0
05A0  CD0001       0243          CALL    DIVIDE        ;CALCULATE D2 RUNNING AVG.
05A3  ED533C08     0244          LD      (D2RAVG),DE    ;STORE AT D2RAVG
05A7  ED5B0F08     0245          LD      DE,(X2ESUM)    ;GET X2ESUM
05AB  2A1108       0246          LD      HL,(X2ESM3)
05AE  2600         0247          LD      H,0
05B0  CD0001       0248          CALL    DIVIDE        ;CALCULATE X2 RUNNING AVG.
05B3  ED533E08     0249          LD      (X2RAVG),DE    ;STORE AT X2RAVG
                   0250 ;
                   0251 ;CHECK IF N FIRINGS REACHED  (J)
                   0252 ;
05B7  AF           0253          XOR     A             ;RESET CARRY FLAG
05B8  2A1708       0254          LD      HL,(NUMFIR)
05BB  ED42         0255          SBC     HL,BC
05BD  C2CB04       0256          JP      NZ,BSYWT      ;WAIT FOR NEXT FIRING UNLESS
                        BC=N
                   0257 ;
                   0258 ;SEND LOW BYTE OF P1 ENERGY AVG. TO PDP-11  (k)
                   0259 ;
05C0  113408       0260 PDPRTN: LD       DE,P1RAVG
05C3  1A           0261          LD      A,(DE)
05C4  D300         0262          OUT     (00H),A
                   0263 ;
                   0264 ;WAIT 12 U-SEC FOR PDP-11 RESPONSE  (l)
                   0265 ;
05C6  0E02         0266          LD      C,2
05C8  0D           0267 LP1:     DEC     C
05C9  CA1B06       0268 LP1A:    JP      Z,DISPLAY
05CC  C3C805       0269 LP1B:    JP      LP1
                   0270 ;
                   0271 ;TRANSFER ENERGY AVG. DATA TO PDP-11  (m)
                   0272 ;
05CF  13           0273 RETRN:   INC     DE            ;HIGH BYTE OF P1 ENERGY
05D0  1A           0274          LD      A,(DE)        ;AVG. TO PDP-11
05D1  D301         0275          OUT     (01H),A
05D3  18FE         0276 H1:      JR      H1
05D5  113608       0277          LD      DE,D1RAVG ;LOW BYTE OF D1 ENERGY
05D8  1A           0278          LD      A,(DE)        ;AVG. SENT TO PDP-11
05D9  D300         0279          OUT     (00H),A
05DB  18FE         0280 H2:      JR      H2
05DD  13           0281          INC     DE            ;HIGH BYTE OF D1 ENERGY
05DE  1A           0282          LD      A,(DE)        ;AVG. SENT TO PDP-11
05DF  D301         0283          OUT     (01H),A
05E1  18FE         0284 H3:      JR      H3
```

Table D1. (continued.)


```
05E3  113808      0285          LD      DE,X1RAVG ;LOW BYTE OF X1 ENERGY
05E6  1A          0286          LD      A,(DE)    ;AVG. SENT TO PDP-11
05E7  D300        0287          OUT     (00H),A
05E9  18FE        0288 H4:      JR      H4
05EB  13          0289          INC     DE        ;HIGH BYTE OF X1 ENERGY
05EC  1A          0290          LD      A,(DE)    ;AVG. SENT TO PDP-11
05ED  D301        0291          OUT     (01H),A
05EF  18FE        0292 H5:      JR      H5
05F1  113A08      0293          LD      DE,P2RAVG ;LOW BYTE OF P2 ENERGY
05F4  1A          0294          LD      A,(DE)    ;AVG. SENT TO PDP-11
05F5  D300        0295          OUT     (00H),A
05F7  18FE        0296 H6:      JR      H6
05F9  13          0297          INC     DE        ;HIGH BYTE OF P2 ENERGY
05FA  1A          0298          LD      A,(DE)    ;AVG. SENT TO PDP-11
05FB  D301        0299          OUT     (01H),A
05FD  18FE        0300 H7:      JR      H7
05FF  113C08      0301          LD      DE,D2RAVG ;LOW BYTE OF D2 ENERGY
0602  1A          0302          LD      A,(DE)    ;AVG. SENT TO PDP-11
0603  D300        0303          OUT     (00H),A
0605  18FE        0304 H8:      JR      H8
0607  13          0305          INC     DE        ;HIGH BYTE OF D2 ENERGY
0608  1A          0306          LD      A,(DE)    ;AVG. SENT TO PDP-11
0609  D301        0307          OUT     (01H),A
060B  18FE        0308 H9:      JR      H9
060D  113E08      0309          LD      DE,X2RAVG ;LOW BYTE OF X2 ENERGY
0610  1A          0310          LD      A,(DE)    ;AVG. SENT TO PDP-11
0611  D300        0311          OUT     (00H),A
0613  18FE        0312 H10:     JR      H10
0615  13          0313          INC     DE        ;HIGH BYTE OF X2 ENERGY
0616  1A          0314          LD      A,(DE)    ;AVG. SENT TO PDP-11
0617  D301        0315          OUT     (01H),A
0619  18FE        0316 H11:     JR      H11
                  0317 ;
                  0318 ;PROGRAM 3-CHANNEL SELECTABLE DISPLAY  (n)
                  0319 ;
061B  DB07        0320 DISPLAY:IN       A,(07H)   ;READ LEFT DISPLAY'S
                  0321                            ;SELECTION SWITCHES
061D  CB9F        0322          RES     3,A
061F  CBA7        0323          RES     4,A
0621  CBAF        0324          RES     5,A
0623  CBB7        0325          RES     6,A
0625  CBBF        0326          RES     7,A
0627  CD7001      0327          CALL    SWSEL1
062A  0E02        0328          LD      C,02H
062C  ED51        0329          OUT     (C),D     ;PROGRAM THOUSANDS AND
                  0330                            ;HUNDREDS OF LEFT DISPLAY
062E  CD8002      0331          CALL    SWSEL2
0631  0E03        0332          LD      C,03H
0633  ED51        0333          OUT     (C),D     ;PROGRAM TENS AND ONES
                  0334                            ;OF LEFT DISPLAY
0635  DB07        0335          IN      A,(07H)   ;READ MIDDLE DISPLAY'S
                  0336                            ;SELECTION SWITCHES
0637  CBB7        0337          RES     6,A
0639  CBBF        0338          RES     7,A
063B  CB3F        0339          SRL     A
063D  CB3F        0340          SRL     A
063F  CB3F        0341          SRL     A
```

Table D1. (concluded.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0007

0641  CD7001      0342          CALL    SWSEL1
0644  0E04        0343          LD      C,04H
0646  ED51        0344          OUT     (C),D       ;PROGRAM THOUSANDS AND
                  0345                              ;HUNDREDS OF MIDDLE DISPLAY
0648  CD8002      0346          CALL    SWSEL2
064B  0E05        0347          LD      C,05H
064D  ED51        0348          OUT     (C),D       ;PROGRAM TENS AND ONES
                  0349                              ;OF MIDDLE DISPLAY
064F  DB0F        0350          IN      A,(0FH)     ;READ RIGHT DISPLAY'S
                  0351                              ;SELECTION SWITCHES
0651  CB9F        0352          RES     3,A
0653  CBA7        0353          RES     4,A
0655  CBAF        0354          RES     5,A
0657  CBB7        0355          RES     6,A
0659  CBBF        0356          RES     7,A
065B  CD7001      0357          CALL    SWSEL1
065E  0E06        0358          LD      C,06H
0660  ED51        0359          OUT     (C),D       ;PROGRAM THOUSANDS AND
                  0360                              ;HUNDREDS OF RIGHT DISPLAY
0662  CD8002      0361          CALL    SWSEL2
0665  0E07        0362          LD      C,07H
0667  ED51        0363          OUT     (C),D       ;PROGRAM TENS AND ONES
                  0364                              ;OF RIGHT DISPLAY
                  0365 ;
                  0366 ;DETERMINE WHERE TO RESUME MAIN PROGRAM  (o)
                  0367 ;
0669  3A1508      0368          LD      A,(NADDR)
066C  CB67        0369          BIT     4,A         ;CHECK IF SWITCHES HAVE BEEN
                  0370                              ;CHANGED BY PDP-11
066E  CA0604      0371          JP      Z,AVGRUN    ;GO TO AVGRUN TO READ SWITCHE
                       S
                  0372                              ;IF SWITCHES UNCHANGED
0671  C31504      0373          JP      CAL         ;OTHERWISE, GO TO SEQUENCE
                  0374                              ;THAT DETERMINES NEW VALUE OF
                  0375                              ;2 RAISED TO N POWER

Errors            0

A.
```

Table D2. Assembled listing of subroutines and service routines.

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;SUBROUTINE REARRG
                    0004 ;==================
                    0005 ;
                    0006 ;
                    0007 ;SUBROUTINE REARRG TAKES THE TWO BYTES
                    0008 ;CONTAINING THE ELEVEN BITS OF A GIVEN
                    0009 ;CHANNEL'S ENERGY FOR A FIRING,FROM THE
                    0010 ;A/D, AND REARRANGES THEM INTO THE
                    0011 ;CORRECT TWO BYTE BINARY NUMBER.
0140  CB42          0012 REARRG: BIT     0,D
0142  CA4A01        0013         JP      Z,L1
0145  CBF3          0014         SET     6,E
0147  C34C01        0015         JP      L2
014A  CBB3          0016 L1:     RES     6,E
014C  CB4A          0017 L2:     BIT     1,D
014E  CA5601        0018         JP      Z,L3
0151  CBFB          0019         SET     7,E
0153  C35801        0020         JP      L4
0156  CBBB          0021 L3:     RES     7,E
0158  CBB2          0022 L4:     RES     6,D
015A  CBBA          0023         RES     7,D
015C  CB3A          0024         SRL     D
015E  CB3A          0025         SRL     D
0160  C9            0026         RET

Errors              0

A.
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;SUBROUTINE ADDREG
                    0004 ;==================
                    0005 ;
                    0006 ;THIS SUBROUTINE ADDS TWO TRIPLE PRECISION NUMBERS.
                    0007 ;BEFORE THIS ROUTINE IS CALLED, THE HL REGISTER
                    0008 ;PAIR MUST POINT TO THE LS-BYTE OF THE TRIPLE
                    0009 ;PRECISION VALUE STORED IN MEMORY.THE LS-BYTE
                    0010 ;OF THE OTHER TRIPLE PRECISION NUMBER MUST BE
                    0011 ;STORED IN REGISTER C AND THE NEXT SIGNIFICANT
                    0012 ;BYTE IN REGISTER D. THE MS-BYTE WILL ALWAYS BE
                    0013 ;00H AND IS TAKEN CARE OF WITHIN THE SUBROUTINE.
                    0014 ;
0120  79            0015 ADDREG: LD      A,C
0121  86            0016         ADD     A,(HL)
0122  77            0017         LD      (HL),A
0123  23            0018         INC     HL
0124  7A            0019         LD      A,D
0125  8E            0020         ADC     A,(HL)
0126  77            0021         LD      (HL),A
0127  23            0022         INC     HL
0128  3E00          0023         LD      A,00H
012A  8E            0024         ADC     A,(HL)
012B  77            0025         LD      (HL),A
012C  C9            0026         RET

Errors              0

A.
```

Table D2. (continued.)


```
                    0001 ;
                    0002 ;
                    0003 ;SUBROUTINE DIVIDE
                    0004 ;==================
                    0005 ;
                    0006 ;THIS SUBROUTINE PERFORMS A 32-BIT BY
                    0007 ;16-BIT UNSIGNED DIVIDE. THE 32-BIT
                    0008 ;DIVIDEND IS INPUT IN H,L,D,AND E
                    0009 ;(H=MSB,E=LSB), WHILE THE 16-BIT DIVISOR
                    0010 ;IS IN REG. PAIR BC. WHEN FINISHED THE
                    0011 ;16-BIT QUOTIENT IS HELD IN DE AND ANY
                    0012 ;REMAINDER IS IN HL.
                    0013 ;
0100  3E10          0014 DIVIDE: LD      A,16        ;ITERATION COUNTER
0102  29            0015 LOOP:   ADD     HL,HL       ;SHIFT HL LEFT
0103  EB            0016         EX      DE,HL       ;DE TO HL FOR SHIFT
0104  29            0017         ADD     HL,HL       ;SHIFT (DE)
0105  EB            0018         EX      DE,HL
0106  D20A01        0019         JP      NC,JUMP1    ;GO IF NO CARRY
0109  23            0020         INC     HL          ;CARRY TO MS 2 BYTES
010A  B7            0021 JUMP1:  OR      A           ;RESET CARRY
010B  ED42          0022         SBC     HL,BC       ;SUBTRACT DIVISOR
010D  13            0023         INC     DE          ;SET Q=1
010E  F21401        0024         JP      P,JUMP2     ;GO IF Q=1
0111  09            0025         ADD     HL,BC       ;RESTORE
0112  CB83          0026         RES     0,E         ;SET Q=0
0114  3D            0027 JUMP2:  DEC     A           ;DECREMENT COUNT
0115  C20201        0028         JP      NZ,LOOP     ;GO IF NOT DONE
0118  C9            0029         RET                 ;RETURN

Errors              0

A.
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;SUBROUTINE SWSEL1
                    0004 ;=================
                    0005 ;
                    0006 ;SUBROUTINE SWSEL1 DETERMINES WHICH CHANNEL
                    0007 ;TO DISPLAY ON THE DISPLAY BEING CONSIDERED,
                    0008 ;CONVERTS THE ENERGY AVERAGE TO BCD, THEN
                    0009 ;PROGRAMS THE THOUSANDS AND HUNDREDS
                    0010 ;DIGITS OF THE DISPLAY
                    0011 ;
      (0349)        0012 THOUSN   EQU     0349H
      (0827)        0013 BINVAL   EQU     0827H
      (0834)        0014 P1RAVG   EQU     834H
      (0836)        0015 D1RAVG   EQU     836H
      (0838)        0016 X1RAVG   EQU     838H
      (083A)        0017 P2RAVG   EQU     83AH
      (083C)        0018 D2RAVG   EQU     83CH
      (083E)        0019 X2RAVG   EQU     83EH
      (0823)        0020 DECML4   EQU     0823H
0170  1600          0021 SWSEL1:  LD      D,00H      ;DISPLAY P1 ENERGY AVG. ?
0172  BA            0022          CP      D
0173  CA9901        0023          JP      Z,S1
0176  1601          0024          LD      D,01H      ;DISPLAY D1 AVG. ?
0178  BA            0025          CP      D
0179  CAA401        0026          JP      Z,S2
017C  1602          0027          LD      D,02H      ;DISPLAY X1 AVG. ?
017E  BA            0028          CP      D
017F  CAAF01        0029          JP      Z,S3
0182  1603          0030          LD      D,03H      ;DISPLAY P2 AVG. ?
0184  BA            0031          CP      D
0185  CABA01        0032          JP      Z,S4
0188  1604          0033          LD      D,04H      ;DISPLAY D2 AVG. ?
018A  BA            0034          CP      D
018B  CAC501        0035          JP      Z,S5
018E  ED5B3E08      0036          LD      DE,(X2RAVG)    ;DISPLAY X2 AVG.
0192  ED532708      0037          LD      (BINVAL),DE
0196  C3CD01        0038          JP      BNTODC
0199  ED5B3408      0039 S1:      LD      DE,(P1RAVG)    ;DISPLAY P1 AVG.
019D  ED532708      0040          LD      (BINVAL),DE
01A1  C3CD01        0041          JP      BNTODC
01A4  ED5B3608      0042 S2:      LD      DE,(D1RAVG)    ;DISPLAY D1 AVG.
01A8  ED532708      0043          LD      (BINVAL),DE
01AC  C3CD01        0044          JP      BNTODC
01AF  ED5B3808      0045 S3:      LD      DE,(X1RAVG)    ;DISPLAY X1 AVG.
01B3  ED532708      0046          LD      (BINVAL),DE
01B7  C3CD01        0047          JP      BNTODC
01BA  ED5B3A08      0048 S4:      LD      DE,(P2RAVG)    ;DISPLAY P2 AVG.
01BE  ED532708      0049          LD      (BINVAL),DE
01C2  C3CD01        0050          JP      BNTODC
01C5  ED5B3C08      0051 S5:      LD      DE,(D2RAVG)    ;DISPLAY D2 AVG.
01C9  ED532708      0052          LD      (BINVAL),DE
01CD  214903        0053 BNTODC:  LD      HL,THOUSN    ;THE ROUTINE BNTODC IS
                    0054                               ;DESCRIBED IN THE
                    0055                               ;Z80 COOKBOOK, PG. 178
01D0  112308        0056          LD      DE,DECML4
01D3  E5            0057 BNDC:    PUSH    HL
```

Table D2. (continued.)


```
01D4  D5          0058          PUSH    DE
01D5  112708      0059 DCEQVL:  LD      DE,BINVAL
01D8  010002      0060          LD      BC,0200H
01DB  A7          0061 DCLOOP:  AND     A
01DC  1A          0062 DCLP1:   LD      A,(DE)
01DD  9E          0063          SBC     A,(HL)
01DE  12          0064          LD      (DE),A
01DF  05          0065          DEC     B
01E0  CAE801      0066          JP      Z,INCRVL
01E3  23          0067          INC     HL
01E4  13          0068          INC     DE
01E5  C3DC01      0069          JP      DCLP1
01E8  0C          0070 INCRVL:  INC     C
01E9  2B          0071          DEC     HL
01EA  112708      0072          LD      DE,BINVAL
01ED  0602        0073          LD      B,02H
01EF  D2DB01      0074          JP      NC,DCLOOP
01F2  0D          0075          DEC     C
01F3  EB          0076          EX      DE,HL
01F4  C5          0077          PUSH    BC
01F5  010200      0078          LD      BC,0002H
01F8  A7          0079 ADDER:   AND     A
01F9  1A          0080 ADDMOR:  LD      A,(DE)
01FA  8E          0081          ADC     A,(HL)
01FB  77          0082          LD      (HL),A
01FC  EDA1        0083          CPI
01FE  E20502      0084          JP      PO,NEXT
0201  13          0085          INC     DE
0202  C3F901      0086          JP      ADDMOR
0205  C1          0087 NEXT:    POP     BC
0206  E1          0088          POP     HL
0207  71          0089          LD      (HL),C
0208  EB          0090          EX      DE,HL
0209  E1          0091          POP     HL
020A  23          0092          INC     HL
020B  23          0093          INC     HL
020C  7B          0094          LD      A,E
020D  FE20        0095          CP      20H
020F  CA1602      0096          JP      Z,A0
0212  1B          0097          DEC     DE
0213  C3D301      0098          JP      BNDC        ;END OF BNTODC
0216  1600        0099 A0:      LD      D,0         ;ARRANGE BITS
0218  3A2308      0100          LD      A,(DECML4)  ;4-7 OF REG. D
                  0101                              ;TO MATCH THE
021B  CB47        0102          BIT     0,A         ;BCD NUMBER AT
021D  CA2202      0103          JP      Z,A1        ;ADDRESS DECML4
0220  CBE2        0104          SET     4,D
0222  CB4F        0105 A1:      BIT     1,A
0224  CA2902      0106          JP      Z,A2
0227  CBEA        0107          SET     5,D
0229  CB57        0108 A2:      BIT     2,A
022B  CA3002      0109          JP      Z,A3
022E  CBF2        0110          SET     6,D
0230  CB5F        0111 A3:      BIT     3,A
0232  CA3702      0112          JP      Z,A4
0235  CBFA        0113          SET     7,D
0237  212308      0114 A4:      LD      HL,DECML4   ;ARRANGE BITS
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0003

023A  2B          0115          DEC     HL          ;0-3 OF REG. D
023B  7E          0116          LD      A,(HL)      ;TO MATCH THE
023C  CB47        0117          BIT     0,A         ;BCD NUMBER AT
023E  CA4302      0118          JP      Z,A5        ;ADDRESS DECML4-1
0241  CBC2        0119          SET     0,D
0243  CB4F        0120 A5:      BIT     1,A
0245  CA4A02      0121          JP      Z,A6
0248  CBCA        0122          SET     1,D
024A  CB57        0123 A6:      BIT     2,A
024C  CA5102      0124          JP      Z,A7
024F  CBD2        0125          SET     2,D
0251  CB5F        0126 A7:      BIT     3,A
0253  CA5802      0127          JP      Z,A8
0256  CBDA        0128          SET     3,D
0258  C9          0129 A8:      RET

Errors            0

A.
```

Table D2. (continued.)

```
;
;
;THOUSN, HUNDRD, TENS, AND ONES TABLES USED IN
;SUBROUTINE SWSEL1
;
ADDRESS                    CONTENTS
-------                    --------

349H              THOUSN:   E8H    ;ONE THOUSAND
34AH                        03H    ;IN BINARY

34BH              HUNDRD:   64H    ;ONE HUNDRED
34CH                        00H    ;IN BINARY

34DH                 TENS:  0AH    ;TEN IN BINARY
34EH                        00H

34FH                 ONES:  01H    ;ONE IN BINARY
350H                        00H

A.
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;SUBROUTINE SWSEL2
                    0004 ;==================
                    0005 ;
                    0006 ;SUBROUTINE SWSEL2 PROGRAMS THE TENS
                    0007 ;AND ONES DIGITS OF THE DISPLAY UNDER
                    0008 ;CONSIDERATION WHEN IT IS CALLED WITH
                    0009 ;THE DATA FOR THE CHANNEL SELECTED IN
                    0010 ;SUBROUTINE SWSEL1.
                    0011 ;
      (0820)        0012 DECMAL  EQU     820H
                    0013 ;
0280  1600          0014 SWSEL2: LD      D,0          ;ARRANGE BITS
0282  3A2008        0015         LD      A,(DECMAL)   ;0-3 OF REG. D
                    0016                              ;TO MATCH THE
0285  CB47          0017         BIT     0,A          ;BCD NUMBER AT
0287  CA8C02        0018         JP      Z,B1         ;ADDRESS DECMAL
028A  CBC2          0019         SET     0,D          ;WHICH HOLDS THE
028C  CB4F          0020 B1:     BIT     1,A          ;ONES DIGIT
028E  CA9302        0021         JP      Z,B2
0291  CBCA          0022         SET     1,D
0293  CB57          0023 B2:     BIT     2,A
0295  CA9A02        0024         JP      Z,B3
0298  CBD2          0025         SET     2,D
029A  CB5F          0026 B3:     BIT     3,A
029C  CAA102        0027         JP      Z,B4
029F  CBDA          0028         SET     3,D
02A1  212008        0029 B4:     LD      HL,DECMAL    ;ARRANGE BITS
02A4  23            0030         INC     HL           ;4-7 OF REG. D
02A5  7E            0031         LD      A,(HL)       ;TO MATCH THE
02A6  CB47          0032         BIT     0,A          ;BCD NUMBER AT
02A8  CAAD02        0033         JP      Z,B5         ;ADDRESS DECMAL+1
02AB  CBE2          0034         SET     4,D          ;WHICH HOLDS THE
02AD  CB4F          0035 B5:     BIT     1,A          ;TENS DIGIT
02AF  CAB402        0036         JP      Z,B6
02B2  CBEA          0037         SET     5,D
02B4  CB57          0038 B6:     BIT     2,A
02B6  CABB02        0039         JP      Z,B7
02B9  CBF2          0040         SET     6,D
02BB  CB5F          0041 B7:     BIT     3,A
02BD  CAC202        0042         JP      Z,B8
02C0  CBFA          0043         SET     7,D
02C2  C9            0044 B8:     RET


Errors              0


A,
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                        0001 ;
                        0002 ;
                        0003 ;INTERRUPT SERVICE ROUTINE PDPSER
                        0004 ;--------------------------------
                        0005 ;
                        0006 ;UPON AN NMI INTERRUPT BY THE PDP-11, THE Z80 INPUTS D
                             TA
                        0007 ;BITS PA0-PA3 FROM THE PDP-11 AND ARRANGES THEM AS THE
                        0008 ;LOWEST ORDER BITS OF AN 8-BIT WORD; THE HIGH ORDER BI
                             S
                        0009 ;ARE ALL RESET. IF BITS PA0-PA3 ARE ALL SET , THEN THE
                        0010 ;PDP-11 HAS NOT CHANGED THE THE VALUE OF N BUT REQUIRE

                        0011 ;THE CURRENT RUNNING AVERAGE FOR EACH CHANNEL. OTHERWI
                             E,
                        0012 ;BITS PA0-PA3 REPRESENT N AND BIT 4 IS SET TO INDICATE
                        0013 ;THAT THE PDP-11 PROVIDES N AS OPPOSED TO THE SWITCHES
                        0014 ;
      (0815)            0015 NADDR   EQU     815H
      (061B)            0016 DISPLAY EQU     61BH
      (05C0)            0017 PDPRTN  EQU     5C0H
                        0018 ;
0329  DB0E              0019 PDPSER: IN      A,(0EH)    ;GET PA0-PA3 FROM PDP-11
032B  CBA7              0020         RES     4,A        ;CLEAR HIGH ORDER BITS
032D  CBAF              0021         RES     5,A
032F  CBB7              0022         RES     6,A
0331  CBBF              0023         RES     7,A
0333  FE0F              0024         CP      0FH
0335  CA4303            0025         JP      Z,NOCHNG   ;LEAVE SERVICE ROUTINE TO OU
                             PUT
                        0026                            ;RUNNING AVERAGES IF N NOT
                        0027                            ;CHANGED BY PDP-11
0338  CBE7              0028         SET     4,A        ;INDICATE N CHANGED BY PDP-1:
033A  321508            0029         LD      (NADDR),A ;STORE NEW N
033D  211B06            0030         LD      HL,DISPLAY;RETURN TO ADDRESS DISPLAY
0340  E3                0031         EX      (SP),HL    ;IN MAIN PROGRAM
0341  ED45              0032         RETN
0343  21C005            0033 NOCHNG: LD      HL,PDPRTN ;RETURN TO ADDRESS PDPRTN
0346  E3                0034         EX      (SP),HL    ;IN MAIN PROGRAM
0347  ED45              0035         RETN


Errors                  0


A.
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;INTERRUPT SERVICE ROUTINE DATRNSFR
                    0004 ;-----------------------------------
                    0005 ;
                    0006 ;THIS INTERRUPT SERVICE ROUTINE HANDLES THE TRANSFER
                          DATA
                    0007 ;FROM THE Z80 TO THE PDP-11 VIA HANDSHAKING.
                    0008 ;
      (05C8)        0009 LP1      EQU     5C8H
      (05C9)        0010 LP1A     EQU     5C9H
      (05CC)        0011 LP1B     EQU     5CCH
      (05CF)        0012 RETRN    EQU     5CFH
                    0013 ;
02E0  C1            0014 TRNSFR:  POP     BC        ;GET CURRENT RETURN ADDRESS
02E1  AF            0015          XOR     A         ;RESET CARRY FLAG
02E2  21C805        0016          LD      HL,LP1
02E5  ED42          0017          SBC     HL,BC     ;USE CURRENT RETURN ADDRESS T
02E7  CA0603        0018          JP      Z,RET2    ;DETERMINE NEW RETURN ADDRESS
02EA  AF            0019          XOR     A
02EB  21C905        0020          LD      HL,LP1A
02EE  ED42          0021          SBC     HL,BC
02F0  CA0603        0022          JP      Z,RET2
02F3  AF            0023          XOR     A
02F4  21CC05        0024          LD      HL,LP1B
02F7  ED42          0025          SBC     HL,BC
02F9  CA0603        0026          JP      Z,RET2
02FC  03            0027 RET1:    INC     BC        ;CHANGE RETURN ADDRESS
02FD  03            0028          INC     BC
02FE  60            0029          LD      H,B
02FF  69            0030          LD      L,C
0300  3B            0031          DEC     SP
0301  3B            0032          DEC     SP
0302  E3            0033          EX      (SP),HL
0303  FB            0034          EI
0304  ED4D          0035          RETI
0306  21CF05        0036 RET2:    LD      HL,RETRN  ;CHANGE RETURN ADDRESS
0309  3B            0037          DEC     SP
030A  3B            0038          DEC     SP
030B  E3            0039          EX      (SP),HL
030C  FB            0040          EI
030D  ED4D          0041          RETI

Errors              0

A.
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;JUMP INSTRUCTION TO BEGINNING OF MAIN PROGRAM
                    0004 ;ON RESET
                    0005 ;
                    0006 ;
      (0400)        0007 MAIN    EQU     400H
                    0008 ;
0000  C30004        0009         JP      MAIN

Errors              0

A.
```

Table D2. (continued.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;JUMP TO INT INTERRUPT SERVICE ROUTINE DATRNSFR
                    0004 ;
                    0005 ;
      (02E0)        0006 DAT     EQU     2E0H
                    0007 ;
0038  C3E002        0008         JP      DAT

Errors              0

A.
```

Table D2. (concluded.)

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001

                    0001 ;
                    0002 ;
                    0003 ;JUMP TO NMI INTERRUPT SERVICE ROUTINE PDPSER
                    0004 ;
      (0329)        0005 PDP     EQU     329H
                    0006 ;
0066  C32903        0007         JP      PDP

Errors              0

A.
```

APPENDIX E

Z80 CPU INSTRUCTION SET*

ALPHABETICAL ASSEMBLY MNEMONIC	OPERATION
ADC HL,ss	Add with Carry Reg. pair ss to HL
ADC A,s	Add with carry operand s to Acc.
ADD A,n	Add value n to Acc.
ADD A,r	Add Reg. r to Acc
ADD A,(HL)	Add location (HL) to Acc.
ADD A,(IX+D)	Add location (IX+d) to Acc.
ADD A,(IY+D)	Add location (IY+d) to Acc.
ADD HL,ss	Add Reg. pair ss to HL
ADD IX,pp	Add Reg. pair pp to IX
ADD IY,rr	Add Reg. pair rr to IY
AND s	Logical 'AND' of operand s and Acc.
BIT b,(HL)	Test BIT b of location (HL)
BIT b,(IX+d)	Test BIT b of location (IX+d)
BIT b,(IY+d)	Test BIT b of location (IY+d)
BIT b,r	Test BIT b or Reg. r
CALL cc,nn	Call subroutine at location nn if condition cc is true
CALL nn	Unconditional call subroutine at location nn
CCF	Complement carry flag
CP s	Compare operand s with Acc.
CPD	Compare location (HL) and Acc. decrement HL
and	and BC
CPDR	Compare location (HL) and Acc. decrement HL and BC, repeat until BC=0
CPI	Compare location (HL) and Acc. increment HL and decrement BC
CPIR	Compare location (HL) and Acc. increment HL, decrement BC repeat until BC=0
CPL	Complement Acc. (1's comp)
DAA	Decimal adjust Acc.

*Reproduced by permission ©1980 Zilog, Inc. This material shall not be reproduced without the written consent of Zilog, Inc.

DEC m	Decrement operand m
DEC IX	Decrement IX
DEC IY	Decrement IY
DEC ss	Decrement Reg. pair ss
DI	Disable interrupts
DJNZ e	Decrement B and Jump relative if B≠0
EI	Enable interrupts
EX (SP),HL	Exchange the location (SP) and HL
EX (SP),IX	Exchange the location (SP) and IX
EX (SP),IY	Exchange the location (SP) and IY
EX AF,AF'	Exchange the contents of AF and AF'
EX DE,HL	Exchange the contents of DE and HL
EXX	Exchange the contents of BC, DE, HL with contents of BC', DE', HL' respectively
HALT	HALT (wait for interrupt or reset)
IM 0	Set interrupt mode 0
IM 1	Set interrupt mode 1
IM 2	Set interrupt mode 2
IN A,(n)	Load the Acc. with input from device n
3 IN r,(C)	Load the Reg. r with input from device (C)
INC (HL)	Increment location (HL)
INC IX	Increment IX
INC (IX+d)	Increment location (IX+d)
INC IY	Increment IY
INC (IY+d)	Increment (IY+d)
INC r	Increment Reg. r
INC ss	Increment Reg. pair ss
IND	Load location (HL) with input from port (C), decrement HL and B
INDR	Load location (HL) with input from port (C), decrement HL and decrement B, repeat until B=0
INI	Load location (HL) with input from port (C); and increment HL and decrement B
INIR	Load location (HL) with input from port (C), increment HL and decrement B, repeat until B=0
JP (HL)	Unconditional Jump to (HL)

JP (IX)	Unconditional Jump to (IX)
JP (IY)	Unconditional Jump to (IY)
JP cc,nn	Jump to location nn if condition cc is true
JP nn	Unconditional jump to location nn
JR C,e	Jump relative to PC+e if carry=1
JR e	Unconditional Jump relative to PC+e
JR NC,e	Jump relative to PC+e if carry=0
JR NZ,e	Jump relative to PC+e if nonzero (Z=0)
JR Z,e	Jump relative to PC+e if zero (Z=1)
LD A,(BC)	Load Acc. with location (BC)
LD A,(DE)	Load Acc. with location (DE)
LD A,I	Load Acc. with I
LD A,(nn)	Load Acc. with location nn
LD A,R	Load Acc. with Reg. R
LD (BC),A	Load location (BC) with Acc.
LD (DE),A	Load location (DE) with Acc.
LD (HL),n	Load location (HL) with value n
LD dd,nn	Load Reg. pair dd with value nn
LD dd,(nn)	Load Reg. pair dd with location (nn)
LD HL,(nn)	Load HL with location (nn)
LD (HL),r	Load location (HL) with Reg. r
LD I,A	Load I with Acc.
LF IX,nn	Load IX with value nn
LD IX,(nn)	Load IX with location (nn)
LD (IX+d),n	Load location (IX+d) with value n
LD (IX+d),r	Load location (IX+d) with Reg. r
LD IY,nn	Load IY with value nn
LD IY,(nn)	Load IY with location (nn)
LD (IY+d),n	Load location (IY+d) with value n
LD (IY+d),r	Load location (IY+d) with Reg. r
LD (nn),A	Load location (nn) with Acc.
LD (nn),dd	Load location (nn) with Reg. pair dd
LD (nn),HL	Load location (nn) with HL
LD (nn),IX	Load location (nn) with IX
LD (nn),IY	Load location (nn) with IY
LD R,A	Load R with Acc.

LD r,(HL)	Load Reg. r with location (HL)
LD r,(IX+d)	Load Reg. r with location (IX+d)
LD r,(IY+d)	Load Reg. r with location (IY+d)
LD r,n	Load Reg. r with value n
LD r,r'	Load Reg. r with Reg. r'
LD SP,HL	Load SP with HL
LD SP,IX	Load SP with IX
LD SP,IY	Load SP with IY
LDD	Load location (DE) with location (HL), decrement DE, HL and BC
LDDR	Load location (DE) with location (HL), decrement DE, HL and BC; repeat until BC=0
LDI	Load location (DE) with location (HL), increment DE, HL, decrement BC
LDIR	Load location (DE) with location (HL), increment DE, HL, decrement BC and repeat until BC=0
NEG	Negate Acc. (2's complement)
NOP	No operation
R s	Logical 'OR' of operand s and Acc.
OTDR	Load output port (C) with location (HL) decrement HL and B, repeat until B=0
OTIR	Load output port (C) with location (HL), increment HL, decrement B, repeat until B=0
OUT (C),r	Load output port (C) with Reg. r
OUT (n),A	Load output port (n) with Acc.
OUTD	Load output port (C) with location (HL), decrement HL and B
OUTI	Load output port (C) with location (HL), increment HL and decrement B
POP IX	Load IX with top of stack
POP IY	Load IY with top of stack
POP qq	Load Reg. pair qq with top of stack
PUSH IX	Load IX onto stack
PUSH IY	Load IY onto stack
PUSH qq	Load Reg. pair qq onto stack

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

This Page Intentionally Left Blank

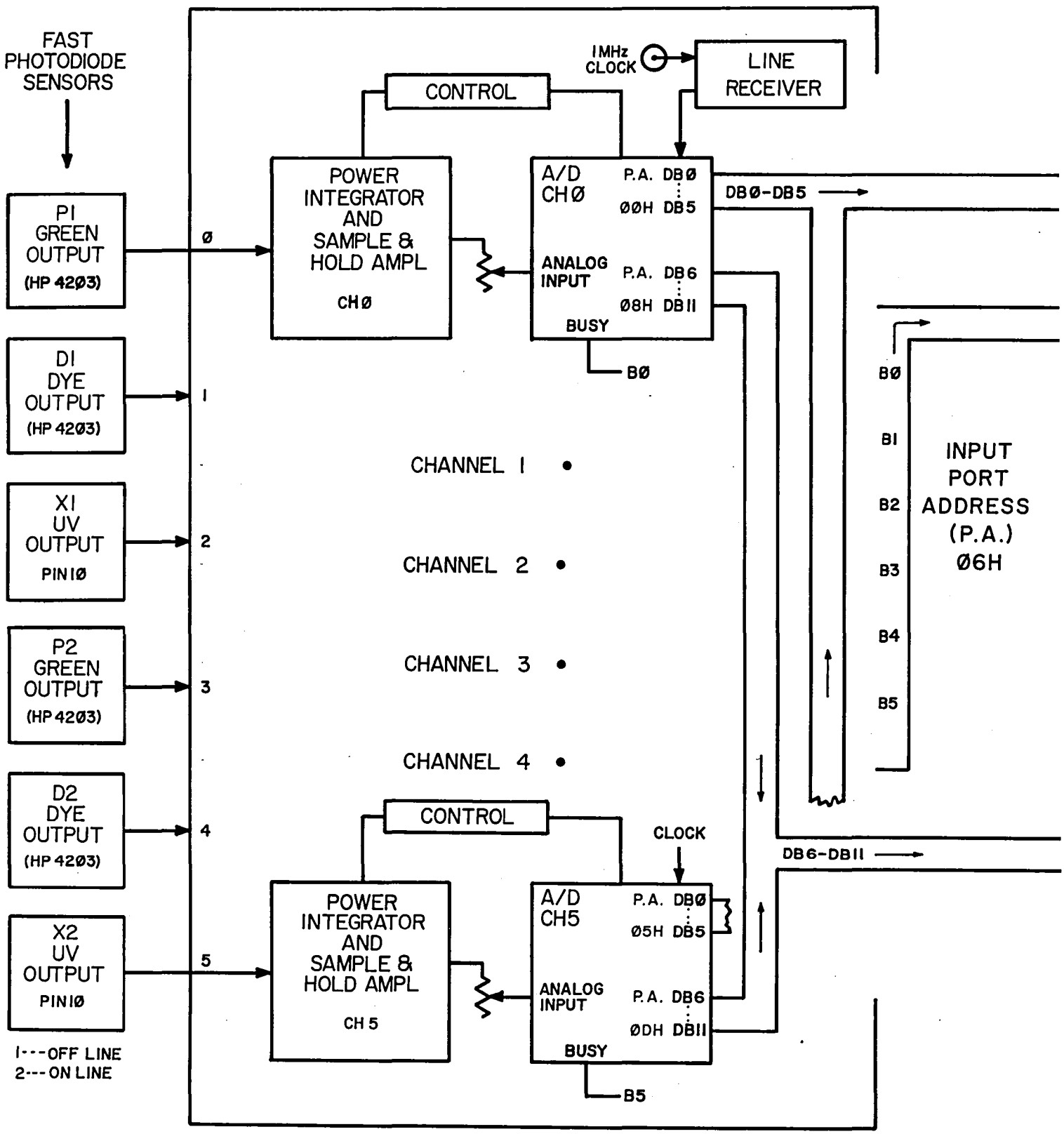
APPENDIX F

COMBINATIONAL BLOCK DIAGRAMS OF SIX-CHANNEL ENERGY MONITORING/DISPLAY SYSTEM AND PMT GAIN CODE MONITOR WITH Z80 MICROCOMPUTER PROCESSING UNIT

This Page Intentionally Left Blank

Figure F1

POWER INTEGRATORS (6-CHANNEL)



This Page Intentionally Left Blank

Figure F2

ENERGY AVERAGING SWITCHES

(LASER ENERGY PER TRANSMISSION AVERAGED OVER 2^N TRANSMISSIONS)

OF TRANSMISSIONS AVERAGED

$2^{0} =$ 1

$2^{2} =$ 4

$2^{4} =$ 16

$2^{6} =$ 64

$2^{8} =$ 256

$2^{10} =$ 1,024

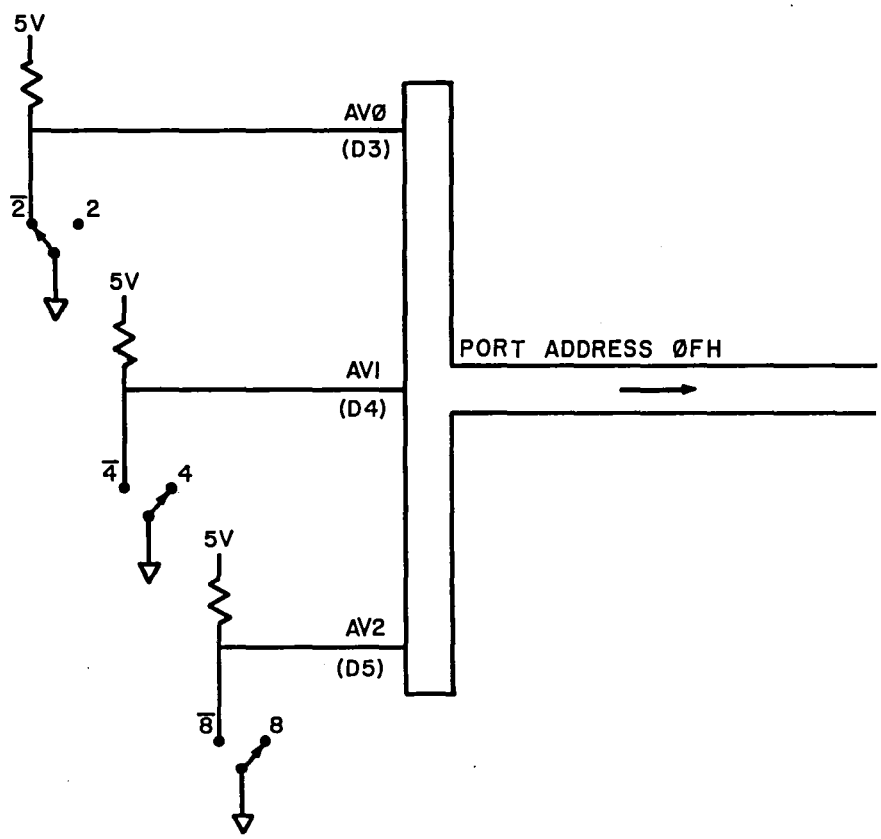
$2^{12} =$ 4,096

$2^{14} =$ 16,384

2^8 2^4 2^2

ENERGY AVERAGING SWITCHES

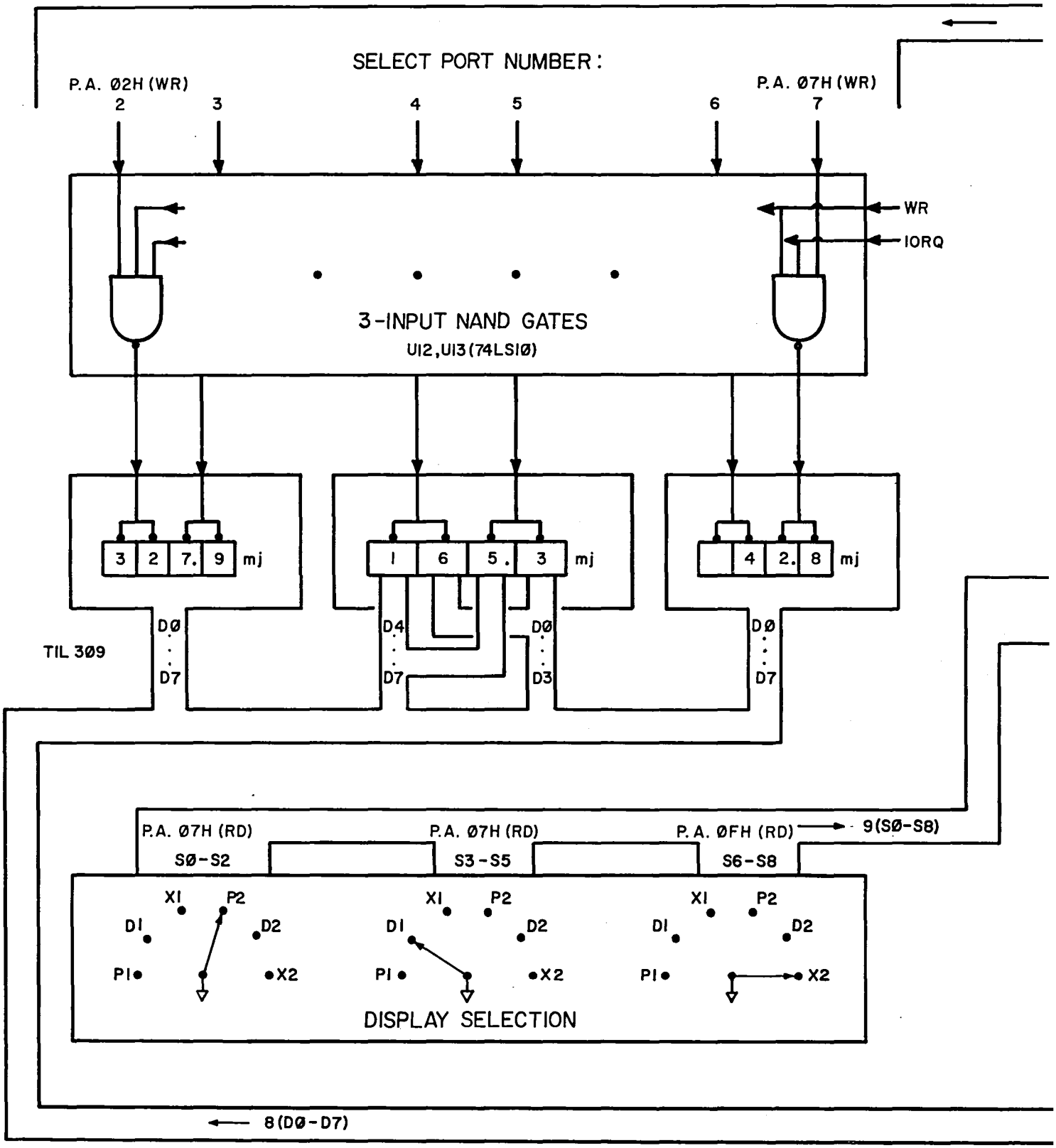
2^N



This Page Intentionally Left Blank

Figure F3

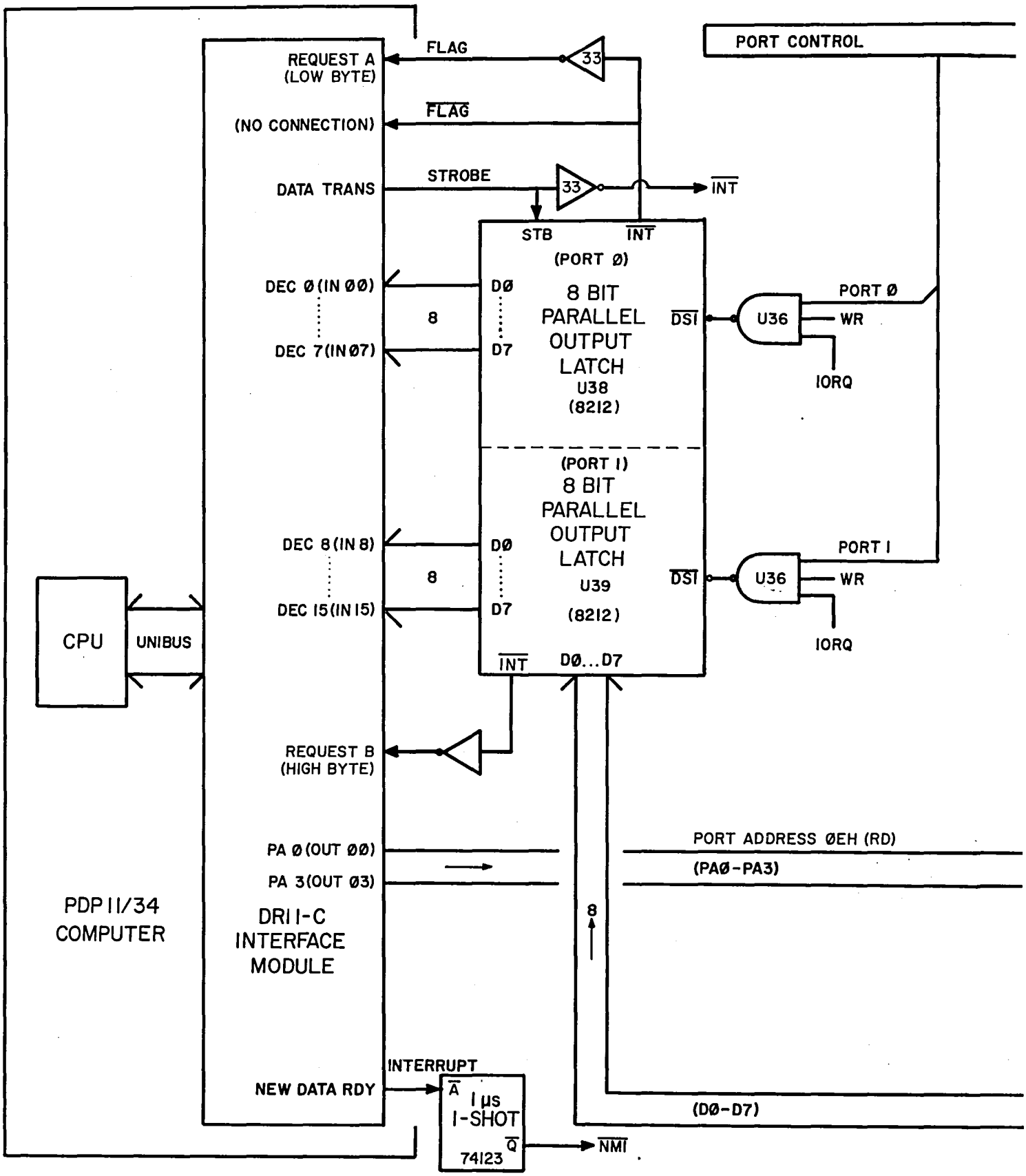
3-CHANNEL SELECTABLE DISPLAY



This Page Intentionally Left Blank

Figure F4

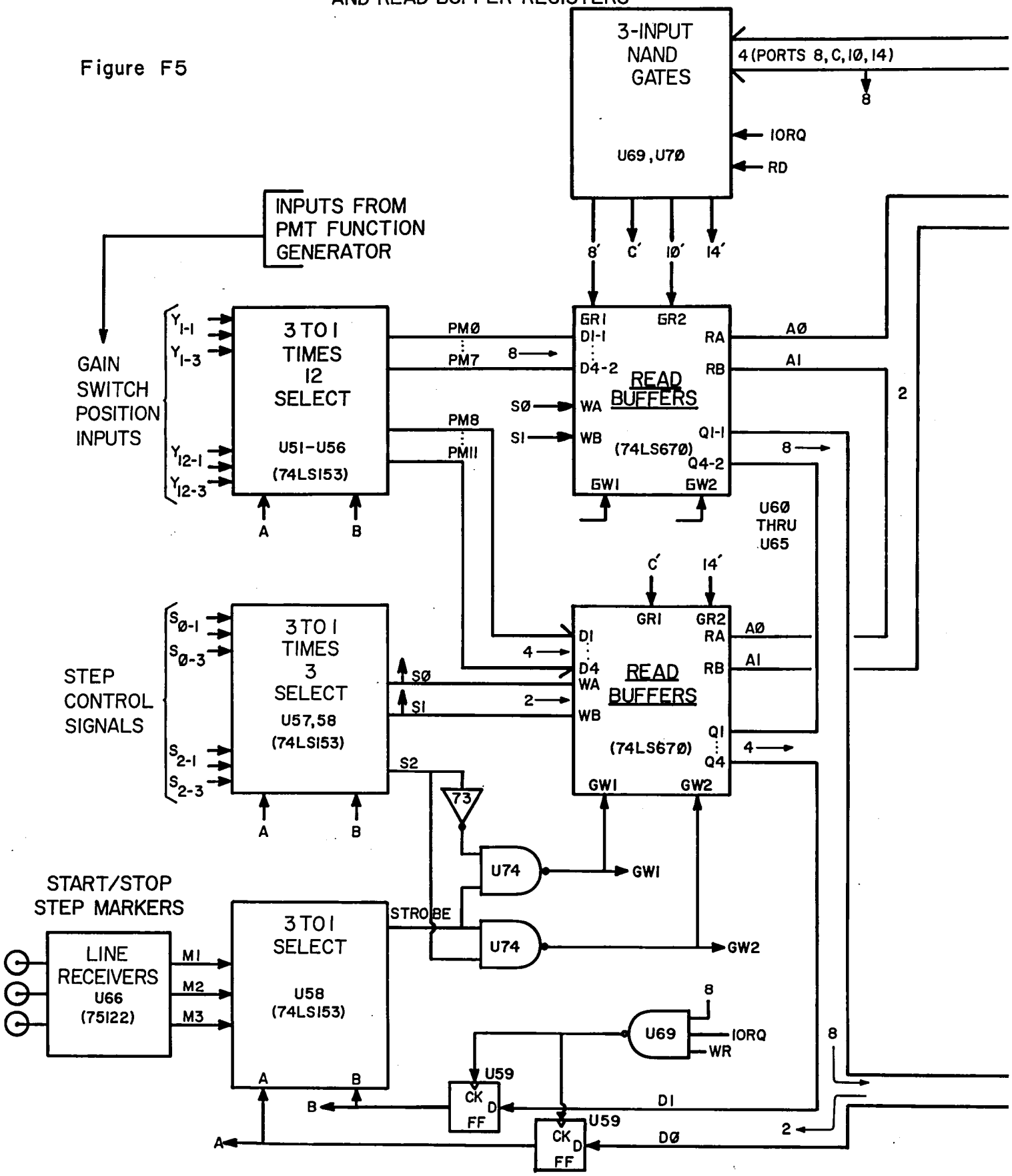
COMMUNICATIONS WITH MAIN COMPUTER



This Page Intentionally Left Blank

STEP GAIN SWITCH CODE MULTIPLEXERS
AND READ BUFFER REGISTERS

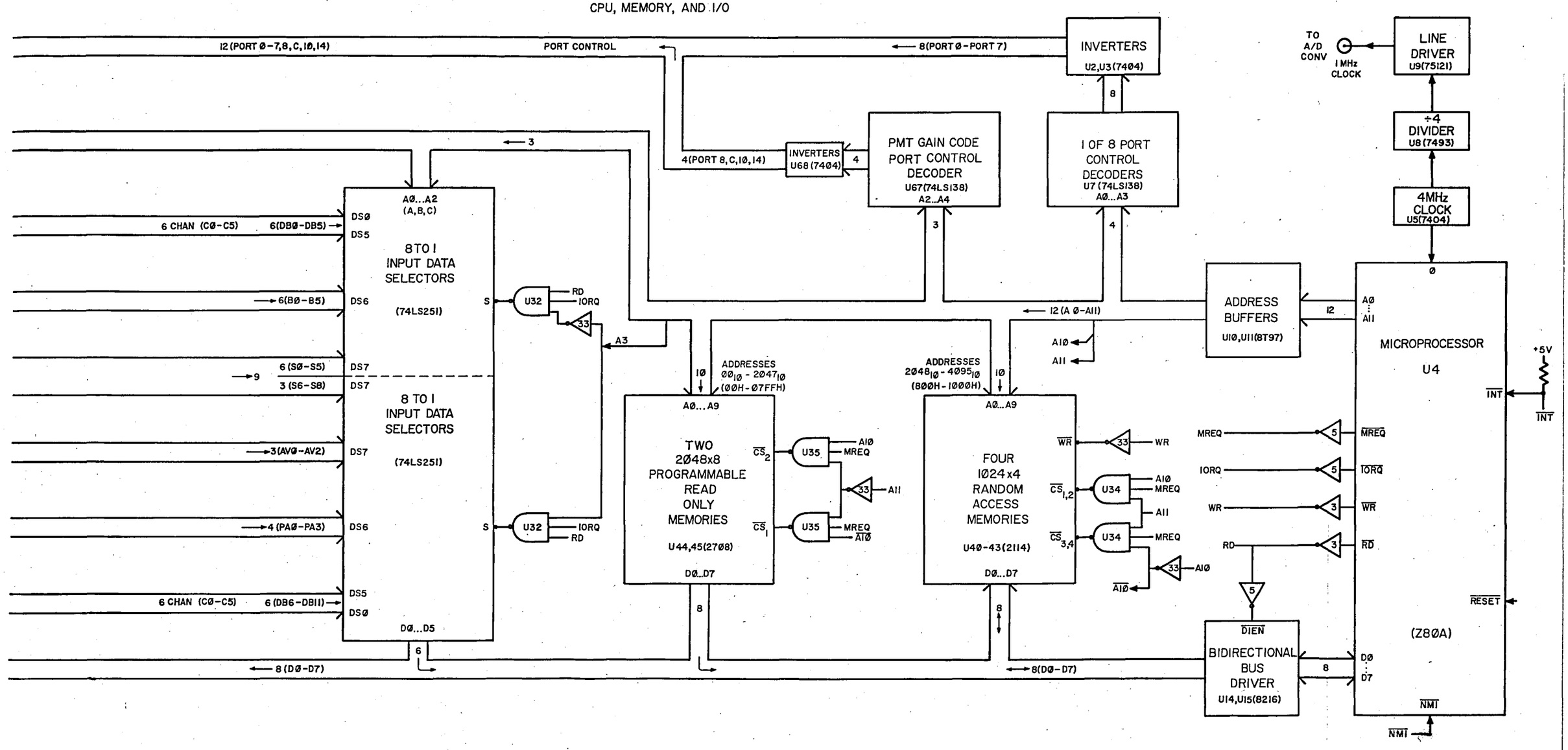
Figure F5



This Page Intentionally Left Blank

Figure F6

CPU, MEMORY, AND I/O



This Page Intentionally Left Blank

REFERENCES

1. Carter, A.F.; Browell, E.V.; Siviter, J.H., Jr; Allen, R.J.; Shipley, S.T.; Butler, C.J.; and Mayo, M.N.: Design and Operation of an Airborne UV DIAL System. Presented at CLEOS (San Diego, Calif.), Feb. 26-28, 1980.

2. Carter, A.F.; Browell, E.V.; Siviter, J.H., Jr; Allen, R.J.; Shipley; S.T.; Butler; C.J.; and Mayo, M.N.: The NASA Langley Multipurpose Airborne DIAL System. Presented at the Tenth International Laser Radar Conference (Silver Springs, Md.), Oct. 6-9, 1980.

3. Butler, C.F.; Shipley, S.T.; and Allen, R.J.: Investigation of Potential of Differential Absorption Lidar Techniques for Remote Sensing of Atmospheric Pollutants. Final Report, NASA Grant NSG-1477, July 1981.

4. Allen, R.J.; and Copeland, G.E.: Multicolor Lidar System Research. Progress Report, NASA Cooperative Agreement NCC1-32, Jan. 1981.

5. Leventhal, L.A.: Z80 Assembly Language Programming. Osborne and Associates, Inc. (Berkeley, Calif.), 1979.

6. Wadsworth, N.: Z80 Gourmet Guide and Cookbook. SCELBI Publications, (Elmwood, Conn.), 1979.

7. Barden, W., Jr.: The Z80 Microcomputer Handbook. Howard W. Sams and Co., Inc. (St. Indianapolis, Ind.), 1979.

8. Z80 Assembly Language Programming. Zilog, Inc. (Cupertino, Calif.), 1980.

9. Cromemco Disk Operating System (CDOS) User's Manual. Cromemco, Inc. (Mountain View, Calif.), Nov. 1978.

10. Cromemco CDOS Test Editor Usesr's Guide. Cromemco, Inc. (Mountain View, Calif.), Aug. 1978.

11. Cromemco Macro Assembler Instruction Manual. Cromemco, Inc. (Mountain View, Calif.), Oct. 1978.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-165875		

4. Title and Subtitle	5. Report Date
PROGRAMMING FOR ENERGY MONITORING/DISPLAY SYSTEM IN MULTICOLOR LIDAR SYSTEM RESEARCH	March 1982
	6. Performing Organization Code
7. Author(s) Ramon C. Alvarado, Jr., Robert J. Allen and Gary E. Copeland	8. Performing Organization Report No. PTR-81-11
	10. Work Unit No.
9. Performing Organization Name and Address Old Dominion University Research Foundation P.O. Box 6369 Norfolk, Virginia 23508-0369	11. Contract or Grant No. NCC1-32
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Research Center Technical Monitor: Dr. E. V. Browell.
Final Report (Programming). Progress Report (Multicolor Lidar System Research).

16. Abstract

This report describes the Z80 microprocessor-based computer program that directs and controls the operation of the six-channel energy monitoring/display system that is a part of the NASA Multipurpose Airborne Differential Absorption Lidar (DIAL) System. The program is written in the Z80 assembly language and is located on EPROM memories. All source and assembled listings of the main program, five subroutines, and two service routines along with flow charts and memory maps are included. A combinational block diagram shows the interfacing (including port addresses) between the six power sensors, displays, front panel controls, the main general purpose minicomputer, and this dedicated microcomputer system.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Energy Monitoring System, microcomputer based. Microcomputer controlled Energy Monitoring System Programming. Optical Laser Energy Monitor System	~~FEDD-Document (For Early Domestic Dissemination)~~ ~~FOREIGN RELEASE ONLY WITH PRIOR NASA APPROVAL & EXPORT LICENSES.~~ ~~Subject Category 61~~

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	126	A06

N-305

For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document