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

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

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 This information may be duplicated and used by the recipient with the express limitation that it not be published.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

Langley Research Center
Hampton, Virginia 23665
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
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>PROGRAM DEVELOPMENT</td>
<td>2</td>
</tr>
<tr>
<td>Objectives</td>
<td>2</td>
</tr>
<tr>
<td>Basic Program Function</td>
<td>3</td>
</tr>
<tr>
<td>Memory Maps</td>
<td>3</td>
</tr>
<tr>
<td>Source Listings of Program Algorithms</td>
<td>6</td>
</tr>
<tr>
<td>Conversion of Assembly Language Program into Machine Language</td>
<td>20</td>
</tr>
<tr>
<td>Program Verification</td>
<td>20</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>22</td>
</tr>
<tr>
<td>APPENDIX A: SYMBOLIC ADDRESSES</td>
<td>23</td>
</tr>
<tr>
<td>APPENDIX B: GENERAL FLOW DIAGRAM</td>
<td>25</td>
</tr>
<tr>
<td>APPENDIX C: PROGRAM FLOW CHARTS AND SOURCE LISTINGS</td>
<td>29</td>
</tr>
<tr>
<td>APPENDIX D: PRINT LISTINGS OF ASSEMBLED PROGRAMS</td>
<td>71</td>
</tr>
<tr>
<td>APPENDIX E: 280 CPU INSTRUCTION SET</td>
<td>93</td>
</tr>
<tr>
<td>APPENDIX F: COMBINATIONAL BLOCK DIAGRAMS OF SIX-CHANNEL ENERGY-MONITORING/DISPLAY SYSTEM AND PMT GAIN CODE MONITOR WITH 280 MICROCOMPUTER PROCESSING UNIT</td>
<td>99</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>113</td>
</tr>
</tbody>
</table>

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Memory map for two 1024 x 8 EPROMS</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Memory map for four 1024 x 4 RAMS</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>List of subroutines and service routines</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Example illustrating program creation</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Example illustrating program conversion from source listing mnemonics into machine language (assembled)</td>
<td>9</td>
</tr>
</tbody>
</table>
LIST OF TABLES - CONCLUDED

Table | Page
-----|------
6    | 13
7    | 14
8    | 18
9    | 19
10   | 21
C1   | 43
C2a  | 52
C2b  | 54
C2c  | 56
C2d  | 61
C2e  | 66
C2f  | 68
C2g  | 70
D1   | 73
D2   | 80

LIST OF FIGURES

Figure | Page
-------|------
1      | 11
2      | 11
3      | 15
4      | 15
## LIST OF FIGURES - CONCLUDED

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>General flow diagram</td>
<td>27</td>
</tr>
<tr>
<td>C1</td>
<td>Flow chart for main program</td>
<td>31</td>
</tr>
<tr>
<td>C2</td>
<td>Flow chart for subroutine REARRG</td>
<td>51</td>
</tr>
<tr>
<td>C3</td>
<td>Flow chart for subroutine ADDREG</td>
<td>53</td>
</tr>
<tr>
<td>C4</td>
<td>Flow chart for subroutine DIVIDE</td>
<td>55</td>
</tr>
<tr>
<td>C5</td>
<td>Flow chart for subroutine SWSEL1</td>
<td>57</td>
</tr>
<tr>
<td>C6</td>
<td>Flow chart for subroutine SWSEL2</td>
<td>64</td>
</tr>
<tr>
<td>C7</td>
<td>Flow chart for service routine PDPSER</td>
<td>67</td>
</tr>
<tr>
<td>C8</td>
<td>Flow chart for service routine DATRNSFR</td>
<td>69</td>
</tr>
<tr>
<td>F1</td>
<td>Power integrators (6-channel)</td>
<td>101</td>
</tr>
<tr>
<td>F2</td>
<td>Energy averaging switches</td>
<td>103</td>
</tr>
<tr>
<td>F3</td>
<td>Three-channel selectable display</td>
<td>105</td>
</tr>
<tr>
<td>F4</td>
<td>Communications with main computer</td>
<td>107</td>
</tr>
<tr>
<td>F5</td>
<td>Step gain switch code multiplexers and read buffer registers</td>
<td>109</td>
</tr>
<tr>
<td>F6</td>
<td>CPU, memory, and I/O</td>
<td>111</td>
</tr>
</tbody>
</table>
PROGRAMMING FOR ENERGY MONITORING/DISPLAY SYSTEM 
IN MULTICOLOR LIDAR SYSTEM RESEARCH 

By 
Ramon C. Alvarado, Jr.\textsuperscript{1}, Robert J. Allen\textsuperscript{2}, 
and Gary E. Copeland\textsuperscript{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\textsuperscript{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 $\mu$m. The dye laser outputs are passed through optical 

\textsuperscript{1}Graduate Research Assistant, Department of Physics, Old Dominion University, Norfolk, VA 23508. 

\textsuperscript{2}Research Professor, Department of Physics, Old Dominion University, Norfolk, VA 23508. 

\textsuperscript{3}Associate Professor, Department of Physics, Old Dominion University, Norfolk, VA 23508. 

\textsuperscript{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 PI 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. Fl). 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. Fl): channel 0 - pump laser PI, 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 ms* 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 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\text{10} (00H - 7FFH), and the RAMs addresses 2048\text{10} to 4095\text{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 $1024 \times 8$ EPROMs.

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>DEC</th>
<th>HEX</th>
<th>BYTES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-20~7</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>JUMP INSTRUCTION TO EXECUTING UF MAIN PROGRAM</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>37</td>
<td>53</td>
<td>NOT USED</td>
</tr>
<tr>
<td>56</td>
<td>38</td>
<td>3A</td>
<td>3</td>
<td>ROUTINE DATAMS</td>
</tr>
<tr>
<td>59</td>
<td>3B</td>
<td>3</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>65</td>
<td>43</td>
<td></td>
<td>NOT USED</td>
</tr>
<tr>
<td>102</td>
<td>66</td>
<td>3</td>
<td>JUMP TO INT INTERRUPT SERVICE</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>68</td>
<td>3</td>
<td>ROUTINE POPSR</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>69</td>
<td>FF</td>
<td>151</td>
<td>NOT Used</td>
</tr>
<tr>
<td>254</td>
<td>100</td>
<td>25</td>
<td>SUBROUTINE DIVIDE</td>
<td></td>
</tr>
<tr>
<td>281</td>
<td>119</td>
<td>7</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>287</td>
<td>11F</td>
<td>13</td>
<td>ROUTINE ADDRES</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>12C</td>
<td>13</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>12D</td>
<td>19</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>319</td>
<td>13F</td>
<td>19</td>
<td>SUBROUTINE REAPPS</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>140</td>
<td>33</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>352</td>
<td>160</td>
<td>39</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>367</td>
<td>16F</td>
<td>15</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>368</td>
<td>170</td>
<td>233</td>
<td>SUBROUTINE SWSER</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>25B</td>
<td>39</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>639</td>
<td>27F</td>
<td>39</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>706</td>
<td>2C2</td>
<td>67</td>
<td>'SUBROUTINE SWSER2</td>
<td></td>
</tr>
<tr>
<td>707</td>
<td>2C3</td>
<td>67</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>735</td>
<td>2DF</td>
<td>29</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>736</td>
<td>2E0</td>
<td>47</td>
<td>SERVICE ROUTINE DATAMS</td>
<td></td>
</tr>
<tr>
<td>782</td>
<td>30E</td>
<td>47</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>320</td>
<td>26</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>809</td>
<td>329</td>
<td>32</td>
<td>SERVICE ROUTINE POPSR</td>
<td></td>
</tr>
<tr>
<td>840</td>
<td>34B</td>
<td>2</td>
<td>THOUSN TABLE</td>
<td></td>
</tr>
<tr>
<td>841</td>
<td>34C</td>
<td>2</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>842</td>
<td>34C</td>
<td>2</td>
<td>HUNDRE TABLE</td>
<td></td>
</tr>
<tr>
<td>843</td>
<td>34E</td>
<td>2</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>844</td>
<td>34E</td>
<td>2</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>845</td>
<td>34F</td>
<td>2</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>846</td>
<td>34F</td>
<td>2</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>847</td>
<td>350</td>
<td>2</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>849</td>
<td>351</td>
<td>175</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>1023</td>
<td>3FF</td>
<td>175</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>400</td>
<td>175</td>
<td>PROGRAM MAIN</td>
<td></td>
</tr>
<tr>
<td>1631</td>
<td>673</td>
<td>628</td>
<td>NOT USED</td>
<td></td>
</tr>
<tr>
<td>1652</td>
<td>674</td>
<td>NOT USED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2047</td>
<td>7FF</td>
<td>396</td>
<td>NOT USED</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th># BYTES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC</td>
<td></td>
<td>PIESUM</td>
</tr>
<tr>
<td>2048</td>
<td>000</td>
<td>PIESUM</td>
</tr>
<tr>
<td>2051</td>
<td>003</td>
<td>DIESUM</td>
</tr>
<tr>
<td>2054</td>
<td>006</td>
<td>XIESUM</td>
</tr>
<tr>
<td>2057</td>
<td>009</td>
<td>P2ESUM</td>
</tr>
<tr>
<td>2060</td>
<td>00C</td>
<td>D2ESUM</td>
</tr>
<tr>
<td>2063</td>
<td>00F</td>
<td>X2ESUM</td>
</tr>
<tr>
<td>2066</td>
<td>012</td>
<td>NOT USED</td>
</tr>
<tr>
<td>2069</td>
<td>015</td>
<td>NOT USED</td>
</tr>
<tr>
<td>2071</td>
<td>017</td>
<td>NUMFIR</td>
</tr>
<tr>
<td>2072</td>
<td>018</td>
<td></td>
</tr>
<tr>
<td>2073</td>
<td>019</td>
<td>NOT USED</td>
</tr>
<tr>
<td>2079</td>
<td>01F</td>
<td></td>
</tr>
<tr>
<td>2080</td>
<td>020</td>
<td>DECIMAL</td>
</tr>
<tr>
<td>2083</td>
<td>023</td>
<td>DECIMAL</td>
</tr>
<tr>
<td>2084</td>
<td>024</td>
<td>NOT USED</td>
</tr>
<tr>
<td>2086</td>
<td>026</td>
<td></td>
</tr>
<tr>
<td>2087</td>
<td>027</td>
<td>PINVAL</td>
</tr>
<tr>
<td>2089</td>
<td>029</td>
<td>NOT USED</td>
</tr>
<tr>
<td>2099</td>
<td>033</td>
<td>NOT USED</td>
</tr>
<tr>
<td>2100</td>
<td>034</td>
<td>PIRAVG</td>
</tr>
<tr>
<td>2101</td>
<td>035</td>
<td></td>
</tr>
<tr>
<td>2102</td>
<td>036</td>
<td>DIRAVG</td>
</tr>
<tr>
<td>2103</td>
<td>037</td>
<td></td>
</tr>
<tr>
<td>2104</td>
<td>038</td>
<td>XIRAVG</td>
</tr>
<tr>
<td>2105</td>
<td>039</td>
<td></td>
</tr>
<tr>
<td>2106</td>
<td>03A</td>
<td>P2RAVG</td>
</tr>
<tr>
<td>2107</td>
<td>03B</td>
<td></td>
</tr>
<tr>
<td>2108</td>
<td>03C</td>
<td>D2RAVG</td>
</tr>
<tr>
<td>2110</td>
<td>03E</td>
<td>X2RAVG</td>
</tr>
<tr>
<td>2112</td>
<td>040</td>
<td>STACK BOTTOM</td>
</tr>
<tr>
<td>2385</td>
<td>951</td>
<td>STACK INITIALIZATION</td>
</tr>
<tr>
<td>2386</td>
<td>952</td>
<td>NOT USED</td>
</tr>
<tr>
<td>4095</td>
<td>FFF</td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>REARRG</td>
<td>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.</td>
</tr>
<tr>
<td>ADDREG</td>
<td>Adds two triple precision numbers.</td>
</tr>
<tr>
<td>DIVIDE</td>
<td>Performs a 32-bit by 16-bit unsigned divide.</td>
</tr>
<tr>
<td>SWSELI1</td>
<td>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.</td>
</tr>
<tr>
<td>SWSEL2</td>
<td>Programs the ten's and one's digits of the display under consideration with the energy data for the channel selected in subroutine SWSELI1.</td>
</tr>
</tbody>
</table>

**SERVICE ROUTINES**

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATRNSFR</td>
<td>Handles the transfer of data from the Z80 to the PDP-11 with handshaking.</td>
</tr>
<tr>
<td>PDPSER</td>
<td>Upon an 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.</td>
</tr>
</tbody>
</table>
Table 4. Example illustrating program creation.

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

A.
A.EDIT ADUREG.Z80

CROMEMCO Text Editor version 00.10

New File

*!

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 ADUREG.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
```

<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode</th>
<th>Mnemonic</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0120</td>
<td>79</td>
<td>ADDREG</td>
<td>LD A,C</td>
</tr>
<tr>
<td>0121</td>
<td>86</td>
<td></td>
<td>ADD A,(HL)</td>
</tr>
<tr>
<td>0122</td>
<td>77</td>
<td></td>
<td>LD (HL),A</td>
</tr>
<tr>
<td>0123</td>
<td>23</td>
<td></td>
<td>INC HL</td>
</tr>
<tr>
<td>0124</td>
<td>7A</td>
<td></td>
<td>LIJ A,[I</td>
</tr>
<tr>
<td>0125</td>
<td>8E</td>
<td></td>
<td>ADC A,(HL)</td>
</tr>
<tr>
<td>0126</td>
<td>77</td>
<td></td>
<td>LD (HL),A</td>
</tr>
<tr>
<td>0127</td>
<td>23</td>
<td></td>
<td>INC HL</td>
</tr>
<tr>
<td>0128</td>
<td>3E00</td>
<td></td>
<td>LD A,O</td>
</tr>
<tr>
<td>012A</td>
<td>8E</td>
<td></td>
<td>ADC A,(HL)</td>
</tr>
<tr>
<td>012C</td>
<td>77</td>
<td></td>
<td>LD (HL),A</td>
</tr>
<tr>
<td>012C</td>
<td>C9</td>
<td></td>
<td>RET</td>
</tr>
</tbody>
</table>

Errors 0
symbols (the blocks and diamonds) of the general flow diagram (fig. Bl) 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 (INT) as well as nonmaskable (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 TNT 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 OFH (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 (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.

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

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).

![Figure 2. Accumulator contents after shifting twice right.](image)

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 (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 AVO to AV2 and PA0 to PA3. Note that the switches, through AVO 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,
Table 6. Data bit configurations that correspond to allowed values of N.

<table>
<thead>
<tr>
<th>D3 AV2, PA3</th>
<th>D2 AV1, PA2</th>
<th>D1 AV0, PA1</th>
<th>D0 0, PA0</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>14</td>
</tr>
</tbody>
</table>
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 DBO 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 DBO 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>
<thead>
<tr>
<th>CHANNEL</th>
<th>LOW BYTE</th>
<th>HIGH BYTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>port 00H</td>
<td>port 08H</td>
</tr>
<tr>
<td>1</td>
<td>port 01H</td>
<td>port 09H</td>
</tr>
<tr>
<td>2</td>
<td>port 02H</td>
<td>port 0AH</td>
</tr>
<tr>
<td>3</td>
<td>port 03H</td>
<td>port 0BH</td>
</tr>
<tr>
<td>4</td>
<td>port 04H</td>
<td>port 0CH</td>
</tr>
<tr>
<td>5</td>
<td>port 05H</td>
<td>port 0DH</td>
</tr>
</tbody>
</table>

*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.

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

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.

![Figure 4. Format of energy data after subroutine REARRG.](image)

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 Pl) is stored in RAM beginning at address PIRAVG (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, B, and D 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 Pl energy average to PDP-ll. - The low byte of the energy average per transmission for channel zero (laser Pl) is output to port 00H, which is one of the two eight-bit parallel output latches connected to the DRll-C interface module (figs. F4 and F6). In addressing this latch, the "REQUEST A" line of the DRll-C interface becomes asserted. This signals the PDP-ll that energy average data is ready to be transferred to it.

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

Transfer of energy average data to PDP-ll. - The high byte of the energy average for channel zero (laser Pl) is output to port 01H, which is
the other eight-bit parallel output latch (figs. F4 and F6). It is impor-
tant 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 aver-
ages 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
OOH, after which the program loops indefinitely until interrupted by a mask-
able interrupt (INT; strobe signal from DR11-C). The high byte is then
output to port O1H, 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 instruc-
tion 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 select-
able display can display the energy averages of any three of the six chan-
nels. 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 dis-
play are S6, S7, and S8, and these are input from port OFH. 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.

<table>
<thead>
<tr>
<th>DATA BYTE</th>
<th>PORT OUTPUT TO</th>
<th>CONDITION FOR NEXT TRANSFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Byte of P1 Avg.</td>
<td>O0H</td>
<td>PDP-11 inputs within 12 μs</td>
</tr>
<tr>
<td>High Byte of P1 Avg.</td>
<td>O1H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>Low Byte of D1 Avg.</td>
<td>O0H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>High Byte of D1 Avg.</td>
<td>O1H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>Low Byte of X1 Avg.</td>
<td>O0H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>High Byte of X1 Avg.</td>
<td>O1H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>Low Byte of P2 Avg.</td>
<td>O0H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>High Byte of P2 Avg.</td>
<td>O1H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>Low Byte of D2 Avg.</td>
<td>O0H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>High Byte of D2 Avg.</td>
<td>O1H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>Low Byte of X2 Avg.</td>
<td>O0H</td>
<td>(INT) via strobe</td>
</tr>
<tr>
<td>High Byte of X2 Avg.</td>
<td>O1H</td>
<td>(INT) via strobe</td>
</tr>
</tbody>
</table>
Table 9. Switch selections for three-channel display.

<table>
<thead>
<tr>
<th>S0, S1, S2</th>
<th>DATA BYTE VALUE</th>
<th>CHANNEL SELECTED FOR DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>or S3, S4, S5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>or S6, S7, S8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0</td>
<td>00H</td>
<td>P1</td>
</tr>
<tr>
<td>0 0 1</td>
<td>01H</td>
<td>D1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>02H</td>
<td>X1</td>
</tr>
<tr>
<td>0 1 1</td>
<td>03H</td>
<td>P2</td>
</tr>
<tr>
<td>1 0 0</td>
<td>04H</td>
<td>D2</td>
</tr>
<tr>
<td>1 0 1</td>
<td>05H</td>
<td>X2</td>
</tr>
</tbody>
</table>

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 SWSELI 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 SWSELI 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 SWSELI 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 NADUR 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 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 OFFFH. 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
-FADDKEX.HEX
-R
NEXT = 0674
-SM800
0800 0800  BB 11
0801 0801  CD 11
0802 0802  01 /
-SB
DE=0000  0FFF
-SB
P=0100 0100  0F4
-SR
A=00  BC=0000  DE=00FF  HL=0000  S=0100  P=04F4  04F4  LD
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
HL,0800  (0B00')
-T
A=00  BC=0000  DE=00FF  HL=0000  S=0100  P=04F7  04F7  LD
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
C,E
-T
A=00  BC=00FF  DE=00FF  HL=0000  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=00FF  HL=0000  S=0100  P=0120  0120  LD
A,C
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,C
-T
A=FF  BC=00FF  DE=00FF  HL=0000  S=00FF  P=0121  0121  ADD
A,(HL)
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-T
H  C  A=10  BC=00FF  DE=00FF  HL=0000  S=00FE  P=0122  0122  LD
(HL),A
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=10  BC=00FF  DE=00FF  HL=0000  S=00FE  P=0123  0123  INC
HL
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=10  BC=00FF  DE=00FF  HL=0001  S=00FE  P=0124  0124  ADD
A,D
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=0F  BC=00FF  DE=00FF  HL=0001  S=00FE  P=0125  0125  ADC
A,(HL)
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=21  BC=00FF  DE=00FF  HL=0001  S=00FE  P=0126  0126  LD
(HL),A
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=21  BC=00FF  DE=00FF  HL=0001  S=00FE  P=0127  0127  INC
HL
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=21  BC=00FF  DE=00FF  HL=0002  S=00FE  P=0128  0128  LD
A,00
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=00  BC=00FF  DE=00FF  HL=0002  S=00FE  P=0129  0129  ADC
A,(HL)
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=01  BC=00FF  DE=00FF  HL=0002  S=00FE  P=012A  012A  LD
(HL),A
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
A,(HL)
-H
C  A=01  BC=00FF  DE=00FF  HL=0002  S=00FE  P=012B  012B  RET
A'=00  B'=0000  D'=0000  H'=0000  X=0000  Y=0000  I=00
-BMB800.S3
0800  10  21  01  -  .1.
OFFH equals 1211OH. A more thorough explanation of DEBUG can be found in reference 11.

ACKNOWLEDGMENTS

This effort was supported by funds provided under NASA Cooperative Agreement NCC1-32 (Dr. E.V. Browell, Technical Monitor). The authors wish to acknowledge the assistance 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 Packaging 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

DECMAL4 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 SWSELI

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

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

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 SWSELI

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 SWSELI
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
START

---

ENABLE INTERRUPTS, CHOOSE INTERRUPT MODE ONE, AND INITIALIZE STACK

(a) INITIALIZE STACK AND SET INTERRUPT MODE

---

P
FROM SECTION (o)

READ SWITCH SETTINGS INTO ACCUMULATOR

(b) READ SWITCHES AND ARRANGE INTO CORRECT N

---

ARRANGE SWITCH SETTINGS INTO NEW N

STORE N IN MEMORY

---

Q
FROM SECTION (o)

CLEAR BIT FOUR OF BYTE FOR N IN ACCUMULATOR

(c) CALCULATE $2^N$ FOR # FIRINGS TO BE AVERAGED

---

CLEAR HL

---

A

Figure C1. Flow chart for main program.
Figure Cl. (continued).
Figure C1. (continued).
Figure C1. (continued).
Figure C1. (continued).
Figure C1. (continued).
INCREMENT SHOT COUNTER (BC)

(h) ADVANCE SHOT COUNTER

CALCULATE P1 RUNNING AVERAGE AND STORE IN MEMORY

(i) CALCULATE RUNNING AVERAGES FOR EACH CHANNEL

CALCULATE D1 RUNNING AVERAGE AND STORE IN MEMORY

CALCULATE X1 RUNNING AVERAGE AND STORE IN MEMORY

CALCULATE P2 RUNNING AVERAGE AND STORE IN MEMORY

CALCULATE D2 RUNNING AVERAGE AND STORE IN MEMORY

CALCULATE X2 RUNNING AVERAGE AND STORE IN MEMORY

Figure C1. (continued).
Figure Cl. (continued).
Figure Cl. (continued).
Figure C1. (continued).
Figure Cl. (continued).
(n) CONTINUED

(a) DETERMINE WHERE TO RESUME MAIN PROGRAM

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
ADDR 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,(OFH) ;READ SWITCH SETTINGS
RES 2,A ;ARRANGE READINGS INTO
RES 4,A ;NEW N
RES 7,A
SRL A
SRL A
LD (NADDR),A ;LOAD ADDRESS HOLDING
;IN WITH NEW N

(continued)
Table C1. (continued.)

; $\text{CALCULATE 2 FOR } \# \text{ FIRINGS TO BE AVERAGED (c)}$

; $\text{CAL: RES 4,A } \# \text{RESET IN CASE SET}$

; $\text{BY PDPSER}$

LD HL,00H \# CLEAR HL
LD D,0 ; N=0?
CP D
JP Z,NO
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
NO: SET 0,L ; N=0, 2 TO N=1
JP STORE
N1: SET 1,L ; N=1, 2 TO N=2
JP STORE
Table Cl. (continued.)

N2: SET 2L ; N=2,2 TO N=4
JP STORE
N3: SET 3L ; N=3,2 TO N=8
JP STORE
N4: SET 4L ; N=4,2 TO N=16
JP STORE
N5: SET 5L ; N=5,2 TO N=32
JP STORE
N6: SET 6L ; N=6,2 TO N=64
JP STORE
N7: SET 7L ; N=7,2 TO N=128
JP STORE
N8: SET 0H ; N=8,2 TO N=256
JP STORE
N9: SET 1H ; N=9,2 TO N=512
JP STORE
N10: SET 2H ; N=10,2 TO N=1024
JP STORE
N11: SET 3H ; N=11,2 TO N=2048
JP STORE
N12: SET 4H ; N=12,2 TO N=4096
JP STORE
N13: SET 5H ; 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/U 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 (g)

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,08H
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)
CALL REARRG
LD HL,X2ESUM
LD CrE
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 CI. (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 Cl. (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
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
ADD REGISTER C TO LOW BYTE OF NUMBER IN MEMORY AND STORE IN SAME MEMORY LOCATION

ADD WITH CARRY REGISTER D TO THE NEXT BYTE OF THE NUMBER IN MEMORY AND STORE IN SAME MEMORY LOCATION

ADD WITH CARRY THE LAST BYTE OF THE NUMBER IN MEMORY TO REGISTER D AND STORE IN THE SAME MEMORY LOCATION

RETURN

Figure C3. Flow chart for subroutine ADDREG.
Table C2b. Subroutine source listing for ADDREG.

```assembly
; SUBROUTINE ADREG
; ===================
; 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 SWSELI.
INITIALIZE POINTERS TO ONE THOUSAND CONSTANT AND DECML4

CALCULATE DECIMAL DIGIT FOR CURRENT POWER OF TEN

SAVE DECIMAL DIGIT IN DECIMAL TABLE

LAST DIGIT CALCULATED?

YES

CLEAR D

LOAD ACCUMULATOR WITH CONTENTS OF ADDRESS DECML4

NO

SET POINTER FOR NEXT POWER OF TEN

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 034H
; D1RAVG EQU 036H
; X1RAVG EQU 038H
; P2RAVG EQU 03AH
; D2RAVG EQU 03CH
; X2RAVG EQU 03EH
; 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 BNOTDC

S1: LD DE,(P1RAVG) ;DISPLAY P1 AVG.
LD (BINVAL),DE
JP BNOTDC

S2: LD DE,(D1RAVG) ;DISPLAY D1 AVG.
LD (BINVAL),DE
JP BNOTDC

S3: LD DE,(X1RAVG) ;DISPLAY X1 AVG.
LD (BINVAL),DE
JP BNOTDC

S4: LD DE,(P2RAVG) ;DISPLAY P2 AVG.
LD (BINVAL),DE
JP BNOTDC
Table C2d. (continued.)

S5:  LD   DE,(D2RAVG)  ; DISPLAY D2 AVG.
     LD   (BINVAL),DE
BNTODC: LD   HL,THOUSN  ; THE ROUTINE BNTODC IS
                ; DESCRIBED IN THE
                ; 80 COOKBOOK, PG. 178
     LD   DE,DECML4
BNDC:  PUSH  HL
       PUSH  DE
DCEQVL: LD   DE,BINVAL
       LD   RC,0200H
DCLOOP: AND   A
DCLP1: LD   A,(DE)
       SBC   A,(HL)
       LD   (DE),A
       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
AO:  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 $A1
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
```
Figure C6. Flow chart for subroutine SWSEL2.
LOAD REGISTER A WITH CONTENTS OF ADDRESS DECIMAL + 1

IS BIT 0 OF REGISTER A SET? YES SET BIT 4 OF REGISTER D

NO

IS BIT 1 OF REGISTER A SET? YES SET BIT 5 OF REGISTER D

NO

IS BIT 2 OF REGISTER A SET? YES SET BIT 6 OF REGISTER D

NO

IS BIT 3 OF REGISTER A SET? YES SET BIT 7 OF REGISTER D

NO

RETURN

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. ; ; DECIMAL EQU 820H

SWSEL2: LD D+0 ;ARRANGE BITS
LD A,(DECIMAL)+0-3 OF REG. D
;TO MATCH THE
BIT 0,A ;BCD NUMBER AT
JP Z,B1 ;ADDRESS DECIMAL
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,DECIMAL ;ARRANGE BITS
INC HL ;4-1 OF REG. D
LD A,(HL) ;TO MATCH THE
BIT 0,A ;BCD NUMBER AT
JP Z,B5 ;ADDRESS DECIMAL+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
Figure C7. Flow chart for service routine PDPSEN.
Table C2f. Subroutine source listing for PDPSER.

; INTERRUPT SERVICE ROUTINE PDPSER
; -----------------------------
; UPON AN NMI INTERRUPT BY THE PDP-11, THE 280 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 5COH
;
; PDPSER: IN A,(OEH) ; 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
POP THE CONTENTS OF THE STACK INTO REGISTER PAIR BC

RESET THE CARRY FLAG

DOES BC = ADDRESS LP1, LP1A, OR LP1B?

YES

RESTORE REGISTER HL AND CHANGE RETURN ADDRESS TO ADDRESS RETURN IN MAIN PROGRAM

RETURN

NO

INCREMENT REGISTER PAIR BC TWICE

LOAD REGISTER PAIR HL WITH CONTENTS OF REGISTER PAIR BC

CHANGE RETURN ADDRESS TO THAT POINTED TO BY REGISTER PAIR HL

RETURN

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 VIA HANDSHAKING.
;
LP1    EQU  5C8H
LP1A   EQU  5C9H
LP1B   EQU  5CCH
RETRN  EQU  5CFH
;
TRANSFR: 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
Table D1. Assembled listing of main program.

CROMEMCO CDOS 280 ASSEMBLER version 02.15

0001 ;
0002 ;
0003 #MAIN PROGRAM LISTING FOR LIDAR ENERGY MONITOR PROGRAM
0004 #***********************************************************************
0005 ;
0006 ;
0007 PIESUM EQU 0800H
0008 DIESUM EQU 0803H
0009 XIESUM EQU 0806H
0010 P2ESUM EQU 0809H
0011 D2ESUM EQU 080CH
0012 X2ESUM EQU 080FH
0013 PIESM3 EQU 0802H
0014 DIESM3 EQU 0805H
0015 XIESM3 EQU 0808H
0016 P2ESM3 EQU 080AH
0017 D2ESM3 EQU 080EH
0018 X2ESM3 EQU 0811H
0019 NADDR EQU 0815H
0020 NUMFIR EQU 0817H
0021 PIRAVG EQU 0819H
0022 DIRAVG EQU 081AH
0023 XIRAVG EQU 081CH
0024 P2RAVG EQU 081DH
0025 D2RAVG EQU 081EH
0026 X2RAVG EQU 081FH
0027 DIVIDE EQU 0100H
0028 REARRG EQU 0140H
0029 ADDRIG EQU 0120H
0030 SWSEL1 EQU 0170H
0031 SWSEL2 EQU 0280H
0032 ;
0033 ;INITIALIZE STACK AND SET INTERRUPT MODE (a)
0034 ;
0035 ;
0036 ;
0037 ;
0038 ;
0039 ;READ SWITCHES AND ARRANGE INTO NEW N (b)
0040 ;
0041 ;
0042 ;
0043 ;
0044 ;
0045 ;
0046 ;
0047 ;
0048 ;
0049 ;
0050 ;CALCULATE 2 FOR 4 FIRINGS TO BE AVERAGED (c)
0051 ;
0052 ;
0053 ;
0054 ;
0055 ;
0056 ;
0057 ;

73
<table>
<thead>
<tr>
<th>Decimal</th>
<th>Hex</th>
<th>Instruction</th>
<th>Address</th>
<th>Label</th>
<th>Mode</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0420</td>
<td>1601</td>
<td>LD D,1</td>
<td>0058</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0422</td>
<td>BA</td>
<td>CP D</td>
<td>0059</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0423</td>
<td>CA7B04</td>
<td>JP Z,N1</td>
<td>0060</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0426</td>
<td>1602</td>
<td>LD D,2</td>
<td>0061</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>0428</td>
<td>BA</td>
<td>CP D</td>
<td>0062</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0429</td>
<td>CA7D04</td>
<td>JP Z,N2</td>
<td>0063</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>042C</td>
<td>1603</td>
<td>LD D,3</td>
<td>0064</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>042E</td>
<td>BA</td>
<td>CP D</td>
<td>0065</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>042F</td>
<td>CA8204</td>
<td>JP Z,N3</td>
<td>0066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0432</td>
<td>1604</td>
<td>LD D,4</td>
<td>0067</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>0434</td>
<td>BA</td>
<td>CP D</td>
<td>0068</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0435</td>
<td>CA8704</td>
<td>JP Z,N4</td>
<td>0069</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0438</td>
<td>1605</td>
<td>LD D,5</td>
<td>0070</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>043A</td>
<td>BA</td>
<td>CP D</td>
<td>0071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>043B</td>
<td>CA8C04</td>
<td>JP Z,N5</td>
<td>0072</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>043E</td>
<td>1606</td>
<td>LD D,6</td>
<td>0073</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>0440</td>
<td>BA</td>
<td>CP D</td>
<td>0074</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0441</td>
<td>CA9104</td>
<td>JP Z,N6</td>
<td>0075</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0444</td>
<td>1607</td>
<td>LD D,7</td>
<td>0076</td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>0446</td>
<td>BA</td>
<td>CP D</td>
<td>0077</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0447</td>
<td>CA9604</td>
<td>JP Z,N7</td>
<td>0078</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>044A</td>
<td>1608</td>
<td>LD D,8</td>
<td>0079</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>044C</td>
<td>BA</td>
<td>CP D</td>
<td>0080</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>044D</td>
<td>CA9B04</td>
<td>JP Z,N8</td>
<td>0081</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0450</td>
<td>1609</td>
<td>LD D,9</td>
<td>0082</td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>0452</td>
<td>BA</td>
<td>CP D</td>
<td>0083</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0453</td>
<td>CAAB04</td>
<td>JP Z,N9</td>
<td>0084</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0456</td>
<td>160A</td>
<td>LD D,OAH</td>
<td>0085</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>0459</td>
<td>BA</td>
<td>CP D</td>
<td>0086</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>045F</td>
<td>CAAD04</td>
<td>JP Z,N11</td>
<td>0087</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0462</td>
<td>160C</td>
<td>LD D,OCH</td>
<td>0091</td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>0464</td>
<td>BA</td>
<td>CP D</td>
<td>0092</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0465</td>
<td>CAAF04</td>
<td>JP Z,N12</td>
<td>0093</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0468</td>
<td>160D</td>
<td>LD D,ODH</td>
<td>0094</td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>046A</td>
<td>BA</td>
<td>CP D</td>
<td>0095</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>046B</td>
<td>CA8404</td>
<td>JP Z,N13</td>
<td>0096</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>046E</td>
<td>CB4F</td>
<td>SET 6,H</td>
<td>0097</td>
<td></td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>0470</td>
<td>C3604</td>
<td>STORE</td>
<td>0098</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0473</td>
<td>CBC5</td>
<td>STORE</td>
<td>0099</td>
<td>N0:</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0475</td>
<td>C36B04</td>
<td>STORE</td>
<td>0100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0478</td>
<td>CBCD</td>
<td>STORE</td>
<td>0101</td>
<td>N1:</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>047A</td>
<td>C36E04</td>
<td>STORE</td>
<td>0102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>047D</td>
<td>CBB5</td>
<td>STORE</td>
<td>0103</td>
<td>N2:</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>047F</td>
<td>C36E04</td>
<td>STORE</td>
<td>0104</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0482</td>
<td>CBDD</td>
<td>STORE</td>
<td>0105</td>
<td>N3:</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>0484</td>
<td>C36F04</td>
<td>STORE</td>
<td>0106</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0487</td>
<td>CBE5</td>
<td>STORE</td>
<td>0107</td>
<td>N4:</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>0489</td>
<td>C36F04</td>
<td>STORE</td>
<td>0108</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>048C</td>
<td>CRED</td>
<td>STORE</td>
<td>0109</td>
<td>N5:</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>048E</td>
<td>C36E04</td>
<td>STORE</td>
<td>0110</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0491</td>
<td>CBF5</td>
<td>STORE</td>
<td>0111</td>
<td>N6:</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>0493</td>
<td>C36E04</td>
<td>STORE</td>
<td>0112</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0496</td>
<td>CBFD</td>
<td>STORE</td>
<td>0113</td>
<td>N7:</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>0498</td>
<td>C36E04</td>
<td>STORE</td>
<td>0114</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table D1. (continued.)

<table>
<thead>
<tr>
<th>CROMEMCO CDOS Z80 ASSEMBLER version 02.15</th>
<th>PAGE 0003</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Hex</th>
<th>Instruction</th>
<th>Parameter(s)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>049B</td>
<td>CBC4</td>
<td>0115 NBI:</td>
<td>SET 0;H</td>
<td>N=0;2 TO N=256</td>
</tr>
<tr>
<td>049D</td>
<td>C3B604</td>
<td>0116</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04A0</td>
<td>C9B5C</td>
<td>0117 N9I:</td>
<td>SET 1;H</td>
<td>N=9;2 TO N=512</td>
</tr>
<tr>
<td>04A2</td>
<td>C9B54</td>
<td>0118</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04A5</td>
<td>C7B604</td>
<td>0119</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04A7</td>
<td>C3B604</td>
<td>0120</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04AC</td>
<td>C3B604</td>
<td>0122</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04AF</td>
<td>CBE4</td>
<td>0123</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04B1</td>
<td>C3B604</td>
<td>0124</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04B4</td>
<td>C9BEC</td>
<td>0125</td>
<td>JP</td>
<td>STORE</td>
</tr>
<tr>
<td>04B6</td>
<td>221708</td>
<td>0126</td>
<td>LD (NUMFIR)+HL</td>
<td>N=0;2 TO N=256</td>
</tr>
<tr>
<td>04B9</td>
<td>0E11</td>
<td>0130</td>
<td>INTRT1:</td>
<td>C,11H</td>
</tr>
<tr>
<td>04BB</td>
<td>A9</td>
<td>0131</td>
<td>XOR A</td>
<td>CLEAR ACCUMULATOR</td>
</tr>
<tr>
<td>04BC</td>
<td>110001</td>
<td>0132</td>
<td>JP</td>
<td>DE,P1ESUM</td>
</tr>
<tr>
<td>04BF</td>
<td>12</td>
<td>0133</td>
<td>JP</td>
<td>DE,P1ESUM</td>
</tr>
<tr>
<td>04C0</td>
<td>B9</td>
<td>0134</td>
<td>CP</td>
<td>C</td>
</tr>
<tr>
<td>04C1</td>
<td>CAC904</td>
<td>0135</td>
<td>JP</td>
<td>Z,J1</td>
</tr>
<tr>
<td>04C4</td>
<td>0D</td>
<td>0136</td>
<td>DEC C</td>
<td>C</td>
</tr>
<tr>
<td>04C5</td>
<td>13</td>
<td>0137</td>
<td>INC DE</td>
<td>C</td>
</tr>
<tr>
<td>04C6</td>
<td>C3BFO4</td>
<td>0139</td>
<td>JP</td>
<td>CLRMEM</td>
</tr>
<tr>
<td>04C9</td>
<td>0600</td>
<td>0140</td>
<td>INITIALIZE SHOT COUNTER</td>
<td>(e)</td>
</tr>
<tr>
<td>04CB</td>
<td>D806</td>
<td>0146</td>
<td>BSYWT:</td>
<td>IN A,(O6H)</td>
</tr>
<tr>
<td>04CD</td>
<td>CB47</td>
<td>0147</td>
<td>BIT 0;A</td>
<td>READ A/D BUSY SIGNALS</td>
</tr>
<tr>
<td>04CF</td>
<td>CAD904</td>
<td>0148</td>
<td>JP</td>
<td>Z,J2</td>
</tr>
<tr>
<td>04D7</td>
<td>CB47</td>
<td>0150</td>
<td>LOOP1:</td>
<td>IN A,(O6H)</td>
</tr>
<tr>
<td>04D8</td>
<td>C2D204</td>
<td>0151</td>
<td>JP</td>
<td>N,Z,LOOP1</td>
</tr>
<tr>
<td>04D9</td>
<td>C3BFO4</td>
<td>0152</td>
<td>JP</td>
<td>J3</td>
</tr>
<tr>
<td>04DC</td>
<td>CB5F</td>
<td>0154</td>
<td>J2:</td>
<td>BIT 3;A</td>
</tr>
<tr>
<td>04DE</td>
<td>CAC904</td>
<td>0155</td>
<td>JP</td>
<td>Z,BSYWT</td>
</tr>
<tr>
<td>04E1</td>
<td>D806</td>
<td>0157</td>
<td>LOOP2:</td>
<td>IN A,(O6H)</td>
</tr>
<tr>
<td>04E3</td>
<td>CB5F</td>
<td>0159</td>
<td>BIT 3;A</td>
<td>LOOP TILL P2 BUSY GOES LOW</td>
</tr>
<tr>
<td>04E5</td>
<td>C2E104</td>
<td>0159</td>
<td>JP</td>
<td>N,Z,LOOP2</td>
</tr>
<tr>
<td>04E8</td>
<td>C5</td>
<td>0163</td>
<td>J3:</td>
<td>PUSH BC</td>
</tr>
<tr>
<td>04E9</td>
<td>0E00</td>
<td>0164</td>
<td>LD C,0</td>
<td>1P1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>04EB</td>
<td>EDS8</td>
<td>0165</td>
<td>IN E,(C)</td>
<td></td>
</tr>
<tr>
<td>04ED</td>
<td>OE08</td>
<td>0166</td>
<td>LD C,0;H</td>
<td></td>
</tr>
<tr>
<td>04EF</td>
<td>ED50</td>
<td>0167</td>
<td>IN D,(C)</td>
<td></td>
</tr>
<tr>
<td>04F1</td>
<td>CD4001</td>
<td>0168</td>
<td>CALL</td>
<td>REARRG</td>
</tr>
<tr>
<td>04F4</td>
<td>210001</td>
<td>0169</td>
<td>LD HL,P1ESUM</td>
<td></td>
</tr>
<tr>
<td>04F7</td>
<td>4B</td>
<td>0170</td>
<td>LD C,E</td>
<td></td>
</tr>
<tr>
<td>04FB</td>
<td>CD2001</td>
<td>0171</td>
<td>CALL</td>
<td>ADDREG</td>
</tr>
</tbody>
</table>

---

(e) INITIALIZE SHOT COUNTER
(f) CHECK IF LASER SYSTEM HAS FIRED
(g) SUM CHANNEL ENERGIES TO PREVIOUS TOTALS
Table D1. (continued.)

CROMEMCO Z80 ASSEMBLER version 02.15

<table>
<thead>
<tr>
<th>Line</th>
<th>Opcode</th>
<th>Address</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04FB</td>
<td>0E01</td>
<td>0172</td>
<td>LD C, 1</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>04FD</td>
<td>ED58</td>
<td>0173</td>
<td>IN E, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>04FF</td>
<td>0E09</td>
<td>0174</td>
<td>LD C, 09H</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0501</td>
<td>ED50</td>
<td>0175</td>
<td>IN D, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0503</td>
<td>CD4001</td>
<td>0176</td>
<td>CALL REARRG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0506</td>
<td>210308</td>
<td>0177</td>
<td>LD HL, #1ESUM</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0509</td>
<td>4B</td>
<td>0178</td>
<td>LD C, E</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>050A</td>
<td>CD2001</td>
<td>0179</td>
<td>CALL ADDRREG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>050D</td>
<td>0E02</td>
<td>0180</td>
<td>LD C, 2</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>050E</td>
<td>ED59</td>
<td>0181</td>
<td>IN E, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0510</td>
<td>0E08</td>
<td>0182</td>
<td>LD C, 08H</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0511</td>
<td>0E0A</td>
<td>0183</td>
<td>IN D, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0515</td>
<td>CD4001</td>
<td>0184</td>
<td>CALL REARRG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0518</td>
<td>210608</td>
<td>0185</td>
<td>LD HL, #1ESUM</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>051A</td>
<td>CD2001</td>
<td>0186</td>
<td>LD C, E</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>051C</td>
<td>0E03</td>
<td>0187</td>
<td>CALL ADDRREG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0521</td>
<td>ED58</td>
<td>0188</td>
<td>IN E, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0523</td>
<td>0E08</td>
<td>0189</td>
<td>LD C, 08H</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0525</td>
<td>ED50</td>
<td>0190</td>
<td>IN D, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0527</td>
<td>CD4001</td>
<td>0191</td>
<td>CALL REARRG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>052A</td>
<td>210908</td>
<td>0192</td>
<td>LD HL, #2ESUM</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>052D</td>
<td>4B</td>
<td>0193</td>
<td>LD C, E</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>052E</td>
<td>CD2001</td>
<td>0194</td>
<td>CALL ADDRREG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0531</td>
<td>0E04</td>
<td>0195</td>
<td>LD C, 4</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0535</td>
<td>ED58</td>
<td>0196</td>
<td>IN E, (C)</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0537</td>
<td>0E0C</td>
<td>0197</td>
<td>LD C, 0CH</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0539</td>
<td>CD4001</td>
<td>0198</td>
<td>CALL REARRG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>053C</td>
<td>210C08</td>
<td>0199</td>
<td>LD HL, #2ESUM</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>053E</td>
<td>4B</td>
<td>0200</td>
<td>LD C, E</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0540</td>
<td>CD2001</td>
<td>0200</td>
<td>CALL ADDRREG</td>
<td>ID1 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0543</td>
<td>0E05</td>
<td>0201</td>
<td>LD C, 5</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0545</td>
<td>ED58</td>
<td>0202</td>
<td>IN E, (C)</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0547</td>
<td>0E0D</td>
<td>0203</td>
<td>LD C, 0DH</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0549</td>
<td>ED50</td>
<td>0204</td>
<td>IN D, (C)</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>054B</td>
<td>CD4001</td>
<td>0205</td>
<td>CALL REARRG</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>054E</td>
<td>210F08</td>
<td>0206</td>
<td>LD HL, #2ESUM</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0551</td>
<td>4B</td>
<td>0207</td>
<td>LD C, E</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0552</td>
<td>CD2001</td>
<td>0208</td>
<td>CALL ADDRREG</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0555</td>
<td>CI</td>
<td>0209</td>
<td>POP BC</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0556</td>
<td>03</td>
<td>0210</td>
<td>INC BC</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0557</td>
<td>ED5B0008</td>
<td>0211</td>
<td>CALL DIVIDE</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0558</td>
<td>2A0208</td>
<td>0212</td>
<td>LD DE, (PIESUM)</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0559</td>
<td>2A0020</td>
<td>0213</td>
<td>LD HL, (PIESUM)</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>055A</td>
<td>2600</td>
<td>0214</td>
<td>LD H, 0</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>055B</td>
<td>CD4001</td>
<td>0215</td>
<td>CALL DIVIDE</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>055C</td>
<td>ED531408</td>
<td>0216</td>
<td>CALL DIVIDE</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>055D</td>
<td>ED5B0308</td>
<td>0217</td>
<td>CALL DIVIDE</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>055E</td>
<td>2A0508</td>
<td>0218</td>
<td>LD DE, (PIESUM)</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>055F</td>
<td>2A0058</td>
<td>0219</td>
<td>LD HL, (PIESUM)</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0560</td>
<td>2600</td>
<td>0220</td>
<td>LD H, 0</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0561</td>
<td>CD4001</td>
<td>0221</td>
<td>CALL DIVIDE</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
<tr>
<td>0570</td>
<td>CD5001</td>
<td>0222</td>
<td>CALL DIVIDE</td>
<td>ID2 ENERGY SUMMING SEQUENCE</td>
</tr>
</tbody>
</table>

; ADVANCE SHOT COUNTER (h)
INC BC

; CALCULATE RUNNING AVERAGES FOR EACH CHANNEL (1)
; CALCULATE P1 RUNNING AVG.
; CALCULATE D1 RUNNING AVG.
Table D1. (continued.)

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

<table>
<thead>
<tr>
<th>Address (Hex)</th>
<th>Instruction(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0573</td>
<td>ED533608</td>
</tr>
<tr>
<td>0577</td>
<td>ED5B0608</td>
</tr>
<tr>
<td>0579</td>
<td>2A0808</td>
</tr>
<tr>
<td>057E</td>
<td>2600</td>
</tr>
<tr>
<td>0580</td>
<td>CD0001</td>
</tr>
<tr>
<td>0583</td>
<td>ED533808</td>
</tr>
<tr>
<td>0587</td>
<td>ED5B0908</td>
</tr>
<tr>
<td>058B</td>
<td>2A0808</td>
</tr>
<tr>
<td>058E</td>
<td>2600</td>
</tr>
<tr>
<td>0590</td>
<td>CD0001</td>
</tr>
<tr>
<td>0593</td>
<td>ED533A08</td>
</tr>
<tr>
<td>0597</td>
<td>ED5B0C08</td>
</tr>
<tr>
<td>059B</td>
<td>2A0E08</td>
</tr>
<tr>
<td>059E</td>
<td>2600</td>
</tr>
<tr>
<td>05A0</td>
<td>CD0001</td>
</tr>
<tr>
<td>05A3</td>
<td>ED533C08</td>
</tr>
<tr>
<td>05A7</td>
<td>ED5B0F08</td>
</tr>
<tr>
<td>05AB</td>
<td>2A1108</td>
</tr>
<tr>
<td>05AE</td>
<td>2600</td>
</tr>
<tr>
<td>05B0</td>
<td>CD0001</td>
</tr>
<tr>
<td>05B3</td>
<td>ED533E08</td>
</tr>
<tr>
<td>05B7</td>
<td>AF</td>
</tr>
<tr>
<td>05B8</td>
<td>2A1708</td>
</tr>
<tr>
<td>05BD</td>
<td>C2CB04</td>
</tr>
<tr>
<td>05C0</td>
<td>113408</td>
</tr>
<tr>
<td>05C3</td>
<td>1A</td>
</tr>
<tr>
<td>05C4</td>
<td>D300</td>
</tr>
<tr>
<td>05C6</td>
<td>0E02</td>
</tr>
<tr>
<td>05CB</td>
<td>0D</td>
</tr>
<tr>
<td>05CC</td>
<td>C3CB05</td>
</tr>
<tr>
<td>05CF</td>
<td>13</td>
</tr>
<tr>
<td>05D0</td>
<td>1A</td>
</tr>
<tr>
<td>05D1</td>
<td>D301</td>
</tr>
<tr>
<td>05D3</td>
<td>18FE</td>
</tr>
<tr>
<td>05D5</td>
<td>113608</td>
</tr>
<tr>
<td>05D8</td>
<td>1A</td>
</tr>
<tr>
<td>05DP</td>
<td>D300</td>
</tr>
<tr>
<td>05DB</td>
<td>18FE</td>
</tr>
<tr>
<td>05DD</td>
<td>13</td>
</tr>
<tr>
<td>05DE</td>
<td>1A</td>
</tr>
<tr>
<td>05DF</td>
<td>D301</td>
</tr>
<tr>
<td>05E1</td>
<td>18FE</td>
</tr>
</tbody>
</table>

BEGIN

<table>
<thead>
<tr>
<th>Instruction(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD (DIRAVG), DE</td>
</tr>
<tr>
<td>LD DE, (X1ESUM)</td>
</tr>
<tr>
<td>LD HL, (X1ESM3)</td>
</tr>
<tr>
<td>CALL DIVIDE</td>
</tr>
<tr>
<td>LD (XIRAVG), DE</td>
</tr>
<tr>
<td>LD DE, (P2ESUM)</td>
</tr>
<tr>
<td>LD HL, (P2ESM3)</td>
</tr>
<tr>
<td>LD H+O</td>
</tr>
<tr>
<td>CALL DIVIDE</td>
</tr>
<tr>
<td>LD (P2RAVG), DE</td>
</tr>
<tr>
<td>LD DE, (D2ESUM)</td>
</tr>
<tr>
<td>LD HL, (D2ESM3)</td>
</tr>
<tr>
<td>LD H+O</td>
</tr>
<tr>
<td>CALL DIVIDE</td>
</tr>
<tr>
<td>LD (D2RAVG), DE</td>
</tr>
<tr>
<td>LD DE, (X2ESUM)</td>
</tr>
<tr>
<td>LD HL, (X2ESM3)</td>
</tr>
<tr>
<td>LD H+O</td>
</tr>
<tr>
<td>CALL DIVIDE</td>
</tr>
<tr>
<td>LD (X2RAVG), DE</td>
</tr>
<tr>
<td>0250 ;</td>
</tr>
</tbody>
</table>
| 0251 ICHECK IF N FIRINGS REACHED (J) ;
| 0252 ;         |
| 0253 XOR A     | RESET CARRY FLAG |
| 0254 LD HL, (NUMFIR) |
| 0255 SBC HL, BC |
| 0256 JP NZ, BSYWT | WAIT FOR NEXT FIRING UNLESS BC=N ;
| 0257 ;         |
| 0258 SEND LOW BYTE OF P1 ENERGY AVG. TO PDP-II (k) ;
| 0259 ;         |
| 0260 PDPRTN: LD DE, P1RAVG |
| 0261 LD A, (DE) |
| 0262 OUT (00H), A |
| 0263 ;         |
| 0264 IWAIT 12 U-SEC FOR PDP-II RESPONSE (1) ;
| 0265 ;         |
| 0266 LD C+2    |
| 0267 LD DEC    |
| 0268 JP Z, DISPLAY |
| 0269 JP LP1    |
| 0270 ;         |
| 0271 TRANSFER ENERGY AVG. DATA TO PDP-II (m) ;
| 0272 ;         |
| 0273 RETRN: INC DE | HIGH BYTE OF P1 ENERGY |
| 0274 LD A, (DE) | IAVG. TO PDP-II |
| 0275 OUT (01H), A |
| 0276 H1: JR H1 |
| 0277 LD DE, DIRAVG | LOW BYTE OF P1 ENERGY |
| 0278 LD A, (DE) | IAVG. SENT TO PDP-II |
| 0279 OUT (00H), A |
| 0280 H2: JR H2 |
| 0281 INC DE | HIGH BYTE OF P1 ENERGY |
| 0282 LD A, (DE) | IAVG. SENT TO PDP-II |
| 0283 OUT (01H), A |
| 0284 H3: JR H3 |

END

77
Table D1. (continued.)

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>05E3</td>
<td>113808 0285 LD</td>
<td>DE·X1AVG LOW BYTE OF X1 ENERGY</td>
</tr>
<tr>
<td>05E6</td>
<td>1A 0286 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>05E7</td>
<td>D300 0287 OUT (00H)A</td>
<td></td>
</tr>
<tr>
<td>05E9</td>
<td>18FE 0288 H4:</td>
<td>JR H4</td>
</tr>
<tr>
<td>05EB</td>
<td>1A 0290 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>05EC</td>
<td>D301 0291 OUT (01H)A</td>
<td></td>
</tr>
<tr>
<td>05EF</td>
<td>18FE 0292 H5:</td>
<td>JR H5</td>
</tr>
<tr>
<td>05F1</td>
<td>113A08 0293 LD</td>
<td>DE·P2AVG LOW BYTE OF P2 ENERGY</td>
</tr>
<tr>
<td>05F4</td>
<td>1A 0294 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>05F5</td>
<td>D300 0295 OUT (00H)A</td>
<td></td>
</tr>
<tr>
<td>05F7</td>
<td>18FE 0296 H6:</td>
<td>JR H6</td>
</tr>
<tr>
<td>05F9</td>
<td>1A 0297 INC DE</td>
<td>HIGH BYTE OF P2 ENERGY</td>
</tr>
<tr>
<td>05FA</td>
<td>1A 0298 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>05FB</td>
<td>D301 0299 OUT (01H)A</td>
<td></td>
</tr>
<tr>
<td>05FD</td>
<td>18FE 0300 H7:</td>
<td>JR H7</td>
</tr>
<tr>
<td>05FF</td>
<td>113C08 0301 LD</td>
<td>DE·D2AVG LOW BYTE OF D2 ENERGY</td>
</tr>
<tr>
<td>0602</td>
<td>1A 0302 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>0603</td>
<td>D300 0303 OUT (00H)A</td>
<td></td>
</tr>
<tr>
<td>0605</td>
<td>18FE 0304 H8:</td>
<td>JR H8</td>
</tr>
<tr>
<td>0607</td>
<td>1A 0305 INC DE</td>
<td>HIGH BYTE OF D2 ENERGY</td>
</tr>
<tr>
<td>060A</td>
<td>1A 0306 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>060B</td>
<td>D301 0307 OUT (01H)A</td>
<td></td>
</tr>
<tr>
<td>060D</td>
<td>113E08 0309 LD</td>
<td>DE·X2AVG LOW BYTE OF X2 ENERGY</td>
</tr>
<tr>
<td>0610</td>
<td>1A 0310 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>0611</td>
<td>D300 0311 OUT (00H)A</td>
<td></td>
</tr>
<tr>
<td>0613</td>
<td>18FE 0312 H10:</td>
<td>JR H10</td>
</tr>
<tr>
<td>0615</td>
<td>1A 0313 INC DE</td>
<td>HIGH BYTE OF X2 ENERGY</td>
</tr>
<tr>
<td>061A</td>
<td>1A 0314 LD A (DE)</td>
<td>AVG. SENT TO PDP-11</td>
</tr>
<tr>
<td>0617</td>
<td>D301 0315 OUT (01H)A</td>
<td></td>
</tr>
<tr>
<td>0619</td>
<td>18FE 0316 H11:</td>
<td>JR H11</td>
</tr>
<tr>
<td>061B</td>
<td>DB07 0320 DISPLAY:</td>
<td>IN A (07H) READ LEFT DISPLAY'S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELECTION SWITCHES</td>
</tr>
<tr>
<td>061D</td>
<td>CB9F 0322 RES 3 A</td>
<td></td>
</tr>
<tr>
<td>061F</td>
<td>CB47 0323 RES 4 A</td>
<td></td>
</tr>
<tr>
<td>0621</td>
<td>CB4F 0324 RES 5 A</td>
<td></td>
</tr>
<tr>
<td>0623</td>
<td>CB7 0325 RES 6 A</td>
<td></td>
</tr>
<tr>
<td>0625</td>
<td>CB8F 0326 RES 7 A</td>
<td></td>
</tr>
<tr>
<td>0627</td>
<td>CD7001 0327 CALL</td>
<td>SWSEL1</td>
</tr>
<tr>
<td>062A</td>
<td>OE02 0328 LD C 02H</td>
<td></td>
</tr>
<tr>
<td>062C</td>
<td>ED51 0329 OUT (C)D</td>
<td>THOUSANDS AND HUNDREDS OF LEFT DISPLAY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>062E</td>
<td>CB8002 0331 CALL</td>
<td>SWSEL2</td>
</tr>
<tr>
<td>0631</td>
<td>OE03 0332 LD C 03H</td>
<td></td>
</tr>
<tr>
<td>0633</td>
<td>ED51 0333 OUT (C)D</td>
<td>TENS AND ONES OF LEFT DISPLAY</td>
</tr>
<tr>
<td>0635</td>
<td>DB07 0335 IN A (07H)</td>
<td>READ MIDDLE DISPLAY'S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELECTION SWITCHES</td>
</tr>
<tr>
<td>0637</td>
<td>CB7 0337 RES 6 A</td>
<td></td>
</tr>
<tr>
<td>0639</td>
<td>CB4F 0338 RES 7 A</td>
<td></td>
</tr>
<tr>
<td>063B</td>
<td>CB3F 0339 SRL 1 A</td>
<td></td>
</tr>
<tr>
<td>063D</td>
<td>CB3F 0340 SRL 1 A</td>
<td></td>
</tr>
<tr>
<td>063F</td>
<td>CB3F 0341 SRL 1 A</td>
<td></td>
</tr>
</tbody>
</table>

78
Table D1. (concluded.)

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

```assembly
0641 CD7001 0342 CALL SWSEL1
0644 OE04 0343 LD C+04H
0646 ED51 0344 OUT (C)+D  #PROGRAM THOUSANDS AND
0648 CD8002 0345 HUNDREDS OF MIDDLE DISPLAY
064B OE05 0346 CALL SWSEL2
064D ED51 0347 LD C+05H
064F DB0F 0348 OUT (C)+D  #OF MIDDLE DISPLAY
0651 CB9F 0349 #PROGRAM TENS AND ONES
0653 CBA7 0350 OF MIDDLE DISPLAY
0655 CB87 0351 #SEQUENCE
0657 CB9F 0352 #THAT DETERMINES
0659 CD7001 0353 NEW VALUE OF
065B OE06 0354 #2 RAISED TO N POWER
0660 ED51 0355 #SELECTION SWITCHES
0662 CD8002 0356 #SELECTION SWITCHES
0665 OE07 0357 #SELECTION SWITCHES
0667 ED51 0358 #SELECTION SWITCHES
0669 3A1508 0359 #SELECTION SWITCHES
066C CB67 0360 #SELECTION SWITCHES
066E CA0604 0361 #SELECTION SWITCHES
0671 C31504 0362 #SELECTION SWITCHES
```

Errors: 0

A.
Table D2. Assembled listing of subroutines and service routines.

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

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 FINISH, FROM THE
0010 ;A/D, AND REARRANGES THEM INTO THE
0011 ;CORRECT TWO BYTE BINARY NUMBER.

0140 CB42 0012 REARRG: BIT 0,0
0142 CA4A01 0013 JP Z,L1
0145 CBF3 0014 SET 6,E
0147 C34C01 0015 JP L2
014A CB63 0016 L1: RES 6,E
014C CB4A 0017 L2: BIT 1,D
014E CA5601 0018 JP 2,L3
0151 CBFB 0019 SET 7,E
0153 C35801 0020 JP L4
0156 CB88 0021 L3: RES 7,E
0158 CB22 0022 L4: RES 6,D
015A CB8A 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

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003</td>
<td>;SUBROUTINE ADDREG</td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td>==============</td>
<td></td>
</tr>
<tr>
<td>0005</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0006</td>
<td>;THIS SUBROUTINE ADDS TWO TRIPLE PRECISION NUMBERS.</td>
<td></td>
</tr>
<tr>
<td>0007</td>
<td>;BEFORE THIS ROUTINE IS CALLED, THE HL REGISTER</td>
<td></td>
</tr>
<tr>
<td>0008</td>
<td>;PAIR MUST POINT TO THE LS-BYTE OF THE TRIPLE</td>
<td></td>
</tr>
<tr>
<td>0009</td>
<td>;PRECISION VALUE STORED IN MEMORY, THE LS-BYTE</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>;OF THE OTHER TRIPLE PRECISION NUMBER MUST BE</td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>;STORED IN REGISTER C AND THE NEXT SIGNIFICANT</td>
<td></td>
</tr>
<tr>
<td>0012</td>
<td>;BYTE IN REGISTER D, THE MS-BYTE WILL ALWAYS BE</td>
<td></td>
</tr>
<tr>
<td>0013</td>
<td>;00H AND IS TAKEN CARE OF WITHIN THE SUBROUTINE.</td>
<td></td>
</tr>
<tr>
<td>0014</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0120 79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0121 86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0122 77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0123 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0124 7A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0125 8E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0126 77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0127 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0128 3E00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>012A 8E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>012B 77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>012C C9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Errors 0

A.
Table D2. (continued.)

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

0001  
0003  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0100  3E10  
0102  29  
0103  EB  
0104  29  
0105  EB  
0106  D20A01  
0109  23  
010A  B7  
010B  ED42  
010D  13  
010E  F21401  
0111  09  
0112  CB83  
0114  3D  
0115  C20201  
0118  C9  

Errors 0

A.
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.

;SWSEL1: LD rl,00H ;DISPLAY P1 ENERGY AVG. ?

CP  D
JMP  L,Sl

LD  D,01H ;DISPLAY U1 AVG. ?

CP  D
JMP  I,S2

LD  D,02H ;DISPLAY XL AVG.

CP  D
JMP  I,S3

LD  D,03H ;DISPLAY P1 AVG. ?

LD  D,04H ;DISPLAY P2 AVG. ?

LD  D,05H ;DISPLAY P2 AVG. ?

LD  D,06H ;DISPLAY P2 AVG. ?

LD  D,07H ;DISPLAY PL AVG.

LD  D,08H ;DISPLAY P2 AVG.

LD  D,09H ;DISPLAY P2 AVG.

LD  D,0AH ;DISPLAY P2 AVG.

LD  D,0BH ;DISPLAY P2 AVG.

LD  D,0CH ;DISPLAY P2 AVG.

LD  D,0DH ;DISPLAY P2 AVG.

LD  D,0EH ;DISPLAY P2 AVG.

LD  D,0FH ;DISPLAY P2 AVG.

LD  D,00H ;DISPLAY P1 AVG. ?

LD  D,01H ;DISPLAY U1 AVG. ?

LD  D,02H ;DISPLAY XL AVG.

LD  D,03H ;DISPLAY P1 AVG. ?

LD  D,04H ;DISPLAY P2 AVG. ?

LD  D,05H ;DISPLAY P2 AVG. ?

LD  D,06H ;DISPLAY P2 AVG. ?

LD  D,07H ;DISPLAY PL AVG.

LD  D,08H ;DISPLAY P2 AVG.

LD  D,09H ;DISPLAY P2 AVG.

LD  D,0AH ;DISPLAY P2 AVG.

LD  D,0BH ;DISPLAY P2 AVG.

LD  D,0CH ;DISPLAY P2 AVG.

LD  D,0DH ;DISPLAY P2 AVG.

LD  D,0EH ;DISPLAY P2 AVG.

LD  D,0FH ;DISPLAY P2 AVG.

LD  D,00H ;DISPLAY P1 AVG. ?

LD  D,01H ;DISPLAY U1 AVG. ?

LD  D,02H ;DISPLAY XL AVG.

LD  D,03H ;DISPLAY P1 AVG. ?

LD  D,04H ;DISPLAY P2 AVG. ?

LD  D,05H ;DISPLAY P2 AVG. ?

LD  D,06H ;DISPLAY P2 AVG. ?

LD  D,07H ;DISPLAY PL AVG.

LD  D,08H ;DISPLAY P2 AVG.

LD  D,09H ;DISPLAY P2 AVG.

LD  D,0AH ;DISPLAY P2 AVG.

LD  D,0BH ;DISPLAY P2 AVG.

LD  D,0CH ;DISPLAY P2 AVG.

LD  D,0DH ;DISPLAY P2 AVG.

LD  D,0EH ;DISPLAY P2 AVG.

LD  D,0FH ;DISPLAY P2 AVG.

LD  D,00H ;DISPLAY P1 AVG. ?

LD  D,01H ;DISPLAY U1 AVG. ?

LD  D,02H ;DISPLAY XL AVG.

LD  D,03H ;DISPLAY P1 AVG. ?

LD  D,04H ;DISPLAY P2 AVG. ?

LD  D,05H ;DISPLAY P2 AVG. ?

LD  D,06H ;DISPLAY P2 AVG. ?

LD  D,07H ;DISPLAY PL AVG.

LD  D,08H ;DISPLAY P2 AVG.

LD  D,09H ;DISPLAY P2 AVG.

LD  D,0AH ;DISPLAY P2 AVG.

LD  D,0BH ;DISPLAY P2 AVG.

LD  D,0CH ;DISPLAY P2 AVG.

LD  D,0DH ;DISPLAY P2 AVG.

LD  D,0EH ;DISPLAY P2 AVG.

LD  D,0FH ;DISPLAY P2 AVG.

LD  D,00H ;DISPLAY P1 AVG. ?

LD  D,01H ;DISPLAY U1 AVG. ?

LD  D,02H ;DISPLAY XL AVG.

LD  D,03H ;DISPLAY P1 AVG. ?

LD  D,04H ;DISPLAY P2 AVG. ?

LD  D,05H ;DISPLAY P2 AVG. ?

LD  D,06H ;DISPLAY P2 AVG. ?

LD  D,07H ;DISPLAY PL AVG.

LD  D,08H ;DISPLAY P2 AVG.

LD  D,09H ;DISPLAY P2 AVG.

LD  D,0AH ;DISPLAY P2 AVG.

LD  D,0BH ;DISPLAY P2 AVG.

LD  D,0CH ;DISPLAY P2 AVG.

LD  D,0DH ;DISPLAY P2 AVG.

LD  D,0EH ;DISPLAY P2 AVG.

LD  D,0FH ;DISPLAY P2 AVG.
Table D2. (continued.)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Opcode</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01D4</td>
<td>D5</td>
<td>0058</td>
<td>PUSH DE</td>
</tr>
<tr>
<td>01D5</td>
<td>112708</td>
<td>0059</td>
<td>DCEVUL: LD DE, BINVAL</td>
</tr>
<tr>
<td>01D8</td>
<td>010002</td>
<td>0060</td>
<td>LD BC,0200H</td>
</tr>
<tr>
<td>01DB</td>
<td>A7</td>
<td>0061</td>
<td>DCLCOP: AND A</td>
</tr>
<tr>
<td>01DC</td>
<td>1A</td>
<td>0062</td>
<td>DCLP1: LD A, (DE)</td>
</tr>
<tr>
<td>01DD</td>
<td>9E</td>
<td>0063</td>
<td>SBC A, (HL)</td>
</tr>
<tr>
<td>01DE</td>
<td>12</td>
<td>0064</td>
<td>LD (DE), A</td>
</tr>
<tr>
<td>01DF</td>
<td>05</td>
<td>0065</td>
<td>DEC B</td>
</tr>
<tr>
<td>01E0</td>
<td>CAE001</td>
<td>0066</td>
<td>JP Z,INCRVL</td>
</tr>
<tr>
<td>01E3</td>
<td>23</td>
<td>0067</td>
<td>INC HL</td>
</tr>
<tr>
<td>01E4</td>
<td>13</td>
<td>0068</td>
<td>INC DE</td>
</tr>
<tr>
<td>01E5</td>
<td>C3DC01</td>
<td>0069</td>
<td>JP DCLP1</td>
</tr>
<tr>
<td>01E8</td>
<td>0C</td>
<td>0070</td>
<td>INCRVL: INC C</td>
</tr>
<tr>
<td>01E9</td>
<td>2B</td>
<td>0071</td>
<td>DEC HL</td>
</tr>
<tr>
<td>01EA</td>
<td>112708</td>
<td>0072</td>
<td>LD DE, BINVAL</td>
</tr>
<tr>
<td>01ED</td>
<td>0602</td>
<td>0073</td>
<td>LD BC,0200H</td>
</tr>
<tr>
<td>01EF</td>
<td>D2D001</td>
<td>0074</td>
<td>JP MC,DCLCOP</td>
</tr>
<tr>
<td>01F2</td>
<td>08</td>
<td>0075</td>
<td>DEC C</td>
</tr>
<tr>
<td>01F3</td>
<td>EB</td>
<td>0076</td>
<td>EX DE,HL</td>
</tr>
<tr>
<td>01F4</td>
<td>C5</td>
<td>0077</td>
<td>PUSH BC</td>
</tr>
<tr>
<td>01F5</td>
<td>010200</td>
<td>0078</td>
<td>LD BC,0002H</td>
</tr>
<tr>
<td>01F8</td>
<td>A7</td>
<td>0079</td>
<td>ADDER: AND A</td>
</tr>
<tr>
<td>01FA</td>
<td>0A</td>
<td>0080</td>
<td>ADDMOR: LD A, (DE)</td>
</tr>
<tr>
<td>01FA</td>
<td>8E</td>
<td>0081</td>
<td>ADC A, (HL)</td>
</tr>
<tr>
<td>01FB</td>
<td>77</td>
<td>0082</td>
<td>LD (HL), A</td>
</tr>
<tr>
<td>01FC</td>
<td>EDA1</td>
<td>0083</td>
<td>CPI</td>
</tr>
<tr>
<td>01FE</td>
<td>E20502</td>
<td>0084</td>
<td>JP POP, NEXT</td>
</tr>
<tr>
<td>0201</td>
<td>13</td>
<td>0085</td>
<td>INC DE</td>
</tr>
<tr>
<td>0202</td>
<td>C2F01</td>
<td>0086</td>
<td>JP ADDMOR</td>
</tr>
<tr>
<td>0205</td>
<td>C1</td>
<td>0087</td>
<td>NEXT: POP BC</td>
</tr>
<tr>
<td>0206</td>
<td>E1</td>
<td>0088</td>
<td>POP HL</td>
</tr>
<tr>
<td>0207</td>
<td>71</td>
<td>0089</td>
<td>LD (HL), C</td>
</tr>
<tr>
<td>0208</td>
<td>EB</td>
<td>0090</td>
<td>EX DE, HL</td>
</tr>
<tr>
<td>0209</td>
<td>E1</td>
<td>0091</td>
<td>POP HL</td>
</tr>
<tr>
<td>020A</td>
<td>23</td>
<td>0092</td>
<td>INC HL</td>
</tr>
<tr>
<td>020B</td>
<td>23</td>
<td>0093</td>
<td>INC HL</td>
</tr>
<tr>
<td>020C</td>
<td>7B</td>
<td>0094</td>
<td>LD A, E</td>
</tr>
<tr>
<td>020D</td>
<td>FC20</td>
<td>0095</td>
<td>CP 20H</td>
</tr>
<tr>
<td>020F</td>
<td>CA1602</td>
<td>0096</td>
<td>JP Z, A0</td>
</tr>
<tr>
<td>0212</td>
<td>1B</td>
<td>0097</td>
<td>DEC DE</td>
</tr>
<tr>
<td>0213</td>
<td>C3D301</td>
<td>0098</td>
<td>JP BND, 0</td>
</tr>
<tr>
<td>0216</td>
<td>1600</td>
<td>0099</td>
<td>AO: LD D, 0</td>
</tr>
<tr>
<td>0219</td>
<td>3A2308</td>
<td>0100</td>
<td>LD A, (DEC4L4), 14-7 UP REG, D</td>
</tr>
<tr>
<td>021B</td>
<td>CB47</td>
<td>0102</td>
<td>BIT 0, A</td>
</tr>
<tr>
<td>021D</td>
<td>CA2202</td>
<td>0103</td>
<td>JP Z, A1</td>
</tr>
<tr>
<td>0220</td>
<td>CB42</td>
<td>0104</td>
<td>SET 4, D</td>
</tr>
<tr>
<td>0222</td>
<td>CB4F</td>
<td>0105</td>
<td>A1: BIT 1, A</td>
</tr>
<tr>
<td>0224</td>
<td>CA2902</td>
<td>0106</td>
<td>JP Z, A2</td>
</tr>
<tr>
<td>0227</td>
<td>CBEA</td>
<td>0107</td>
<td>SET 5, D</td>
</tr>
<tr>
<td>0229</td>
<td>CB57</td>
<td>0108</td>
<td>A2: BIT 2, A</td>
</tr>
<tr>
<td>022B</td>
<td>CA3002</td>
<td>0109</td>
<td>JP Z, A3</td>
</tr>
<tr>
<td>022E</td>
<td>CBF2</td>
<td>0110</td>
<td>SET 6, D</td>
</tr>
<tr>
<td>0230</td>
<td>CB5F</td>
<td>0111</td>
<td>A3: BIT 3, A</td>
</tr>
<tr>
<td>0232</td>
<td>CA3702</td>
<td>0112</td>
<td>JP Z, A4</td>
</tr>
<tr>
<td>0235</td>
<td>CBFA</td>
<td>0113</td>
<td>SET 7, D</td>
</tr>
<tr>
<td>0237</td>
<td>212308</td>
<td>0114</td>
<td>AO: LD ML, (DEC4L4), 14-7 UP REG, D</td>
</tr>
</tbody>
</table>
Table D2. (continued.)

<table>
<thead>
<tr>
<th>Address</th>
<th>Code</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>023A 2B</td>
<td>0115</td>
<td>DEC HL</td>
<td>0-3 OF REG. D</td>
</tr>
<tr>
<td>023B 7E</td>
<td>0116</td>
<td>LD A (HL)</td>
<td>TO MATCH THE</td>
</tr>
<tr>
<td>023C C8</td>
<td>0117</td>
<td>BIT 0, A</td>
<td>FCU NUMBER A1</td>
</tr>
<tr>
<td>023E CA</td>
<td>0118</td>
<td>JP Z, A5</td>
<td>ADDRESS DECML 4-1</td>
</tr>
<tr>
<td>0241 C8</td>
<td>0119</td>
<td>SET 0, D</td>
<td></td>
</tr>
<tr>
<td>0243 CB</td>
<td>0120</td>
<td>BIT 1, A</td>
<td></td>
</tr>
<tr>
<td>0245 CA</td>
<td>0121</td>
<td>JP Z, A6</td>
<td></td>
</tr>
<tr>
<td>0248 CB</td>
<td>0122</td>
<td>SET 1, D</td>
<td></td>
</tr>
<tr>
<td>024A CB</td>
<td>0123</td>
<td>BIT 2, A</td>
<td></td>
</tr>
<tr>
<td>024C CA</td>
<td>0124</td>
<td>JP Z, A7</td>
<td></td>
</tr>
<tr>
<td>024F CB</td>
<td>0125</td>
<td>SET 2, D</td>
<td></td>
</tr>
<tr>
<td>0251 CB</td>
<td>0126</td>
<td>BIT 3, A</td>
<td></td>
</tr>
<tr>
<td>0253 CA</td>
<td>0127</td>
<td>JP Z, A8</td>
<td></td>
</tr>
<tr>
<td>0256 CB</td>
<td>0128</td>
<td>SET 3, D</td>
<td></td>
</tr>
</tbody>
</table>

Errors: 0

A.
Table D2. (continued.)

; THOUSN, HUNDRED, TENS, AND ONES TABLES USED IN SUBROUTINE SWSEL1

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>349H</td>
<td>THOUSN: EBH</td>
</tr>
<tr>
<td>34AH</td>
<td></td>
</tr>
<tr>
<td>34BH</td>
<td>HUNDRED: 64H</td>
</tr>
<tr>
<td>34CH</td>
<td></td>
</tr>
<tr>
<td>34DH</td>
<td>TENS: 0AH</td>
</tr>
<tr>
<td>34EH</td>
<td></td>
</tr>
<tr>
<td>34FH</td>
<td>ONES: 01H</td>
</tr>
<tr>
<td>350H</td>
<td></td>
</tr>
</tbody>
</table>

A.
Table D2. (continued.)

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

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 ;
(O820)
0012 DECIMAL EQU B20H
0013 ;
0280 1600 0014 SWSEL2: LD D,0 ;ARRANGE BITS
0282 3A2008 0015 LD A,(DECIMAL)+0-3 OF REG. D
0285 CB47 0016 J0 MATCH THE
0287 CA8C02 0017 BIT 0,A ;BCD NUMBER A1
028A CBC2 0018 JP Z,B1 ;ADDRESS DECIMAL
028C CB4F 0019 SET 0,D ;WHICH HOLDS THE
028E CA9302 0020 B1: BIT 1,A ;ONES DIGIT
0291 CBCA 0021 JP Z,B2
0293 CB57 0022 SET 1,D
0295 CA9A02 0023 B2: BIT 2,A
0298 CB02 0024 JP Z,B3
029A CB5F 0025 SET 2,D
029C CA102 0026 B3: BIT 3,A
029F CB0A 0027 JP Z,B4
02A1 212008 0028 SET 3,D
02A4 23 0029 B4: LD HL,DECIMAL ;ARRANGE BITS
02A5 7E 0030 INC HL ;4-7 OF REG. D
02A6 CB47 0031 LD A,(HL) ;TU MATCH THE
02A8 CA9D02 0032 BIT 0,A ;BCD NUMBER A1
02AB CBE2 0033 JP Z,B5 ;ADDRESS DECIMAL+1
02AD CB4F 0034 SET 4,D ;WHICH HOLDS THE
02AF CAB402 0035 B5: BIT 1,A ;ONES DIGIT
02B2 CBEA 0036 JP Z,B6
02B4 CB57 0037 SET 5,D
02B6 CABB02 0038 B6: BIT 2,A
02B9 CBF2 0039 JP Z,B7
02BB CB5F 0040 SET 6,D
02BD CA2C02 0041 B7: BIT 3,A
02CC CB08 0042 JP Z,B8
02C0 C9 0043 SET 7,D
02C2 CBFA 0044 B8: RET

Errors 0

A.
Table D2. (continued.)

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

0001 ;
0002 ;
0003 ;INTERRUPT SERVICE ROUTINE PDPRTN
0004 ;---------------------------------
0005 ;
0006 ;UPON AN NMI INTERRUPT BY THE PDP-11, THE Z80 INPUTS DATA
0007 ;BITS PAO-PA3 FROM THE PDP-11 AND ARRANGES THEM AS THE
0008 ;LOWEST ORDER BITS OF AN 8-BIT WORD; THE HIGH ORDER BITS
0009 ;ARE ALL RESET. IF BITS PAO-PA3 ARE ALL SET, THEN THE
0010 ;PDP-11 HAS NOT CHANGED THE VALUE OF N BUT REQUIRE
0011 ;THE CURRENT RUNNING AVERAGE FOR EACH CHANNEL. OTHERWIS
0012 ;BITS PAO-PA3 REPRESENT N AND BIT 4 IS SET TO INDICATE
0013 ; THAT THE PDP-11 PROVIDES N AS OPPOSED TO THE SWITCHES
0014 ;
0015 (0815) 0016 NA DDR EQU 815H
0017 (061B) 0018 DISPLAY EQU 61BH
0019 (05C0) 0020 PDPRTN EQU 5C0H
0021 ;
0022 0329 DBOE 0023 PDPRTN: IN A,(0EH) ;GET PAO-PA3 FROM PDP-11
0024 032B CBA7 0025 RES 4,A ;CLEAR HIGH ORDER BITS
0026 032D CBAF 0027 RES 5,A
0028 032F CBF7 0029 RES 6,A
0029 0331 CBBF 0030 RLS 7,A
0030 0333 FE0F 0031 CP 0FH
0031 0335 CA4303 0032 JP Z,NOCHNG ;LEAVE SERVICE ROUTINE TO QU
0032 PUT
0033 ;RUNNING AVERAGES IF N NOT
0034 0026 ;CHANGED BY PDP-11
0035 0027 ;
0036 0028 0338 CBE7 0029 SET 4,A ;INDICATE N CHANGED BY PDP-11:
0037 0030 033A 321508 0031 LD (NADDR),A ;STORE NEW N
0038 0032 033D 211B06 0033 LD HL,D 0034 RETN
0034 ;DISPLAY;RETURN TO ADDRESS DISPLAY
0035 0035 0340 E3 0036 (SP),HL ;IN MAIN PROGRAM
0036 0037 0341 ED45 0038 RETN
0037 0038 0342 21C005 0039 NOCHNG: LD HL,PDPRTN ;RETURN TO ADDRESS PDPRTN
0039 0040 0343 E3 0041 (SP),HL ;IN MAIN PROGRAM
0041 0042 0346 ED45 0043 EX
0043 0044 0347 0044 RETN

Errors 0

A.
Table D2. (continued.)

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td></td>
<td>INTERRUPT SERVICE ROUTINE TRANSFER</td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td>THIS INTERRUPT SERVICE ROUTINE HANDLES THE TRANSFER</td>
</tr>
<tr>
<td>0003</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>0004</td>
<td></td>
<td>FROM THE Z80 TO THE PDP-11 VIA HANDSHAKING</td>
</tr>
<tr>
<td>(05C8)</td>
<td>0009</td>
<td>LPI EOU 5C8H</td>
</tr>
<tr>
<td>(05C9)</td>
<td>0010</td>
<td>LPIA EOU 5C9H</td>
</tr>
<tr>
<td>(05CC)</td>
<td>0011</td>
<td>LP18 EOU 5CCH</td>
</tr>
<tr>
<td>(05CF)</td>
<td>0012</td>
<td>RETRN EOU 5CFH</td>
</tr>
<tr>
<td>0013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02E0 C1</td>
<td>0014</td>
<td>TRNSFR: POP BC GET CURRENT RETURN ADDRESS</td>
</tr>
<tr>
<td>02E1 AF</td>
<td>0015</td>
<td>XOR A RESET CARRY FLAG</td>
</tr>
<tr>
<td>02E2 21C805</td>
<td>0016</td>
<td>LD HL+LP1</td>
</tr>
<tr>
<td>02E5 ED42</td>
<td>0017</td>
<td>SBC HL+BC USE CURRENT RETURN ADDRESS</td>
</tr>
<tr>
<td>02E7 CA0603</td>
<td>0018</td>
<td>JP Z+RET2 DETERMINE NEW RETURN ADDRESS</td>
</tr>
<tr>
<td>02EA AF</td>
<td>0019</td>
<td>XOR A</td>
</tr>
<tr>
<td>02EB 21C905</td>
<td>0020</td>
<td>LD HL+LP1A</td>
</tr>
<tr>
<td>02EE ED42</td>
<td>0021</td>
<td>SBC HL+BC</td>
</tr>
<tr>
<td>02F0 CA0603</td>
<td>0022</td>
<td>JP Z+RET2</td>
</tr>
<tr>
<td>02F3 AF</td>
<td>0023</td>
<td>XOR A</td>
</tr>
<tr>
<td>02F4 21CC05</td>
<td>0024</td>
<td>LD HL+LP1B</td>
</tr>
<tr>
<td>02F7 ED42</td>
<td>0025</td>
<td>SBC HL+BC</td>
</tr>
<tr>
<td>02F9 CA0603</td>
<td>0026</td>
<td>JP Z+RET2</td>
</tr>
<tr>
<td>02FC 03</td>
<td>0027</td>
<td>RET1: INC BC CHANGE RETURN ADDRESS</td>
</tr>
<tr>
<td>02FD 03</td>
<td>0028</td>
<td>INC HL</td>
</tr>
<tr>
<td>02FE 40</td>
<td>0029</td>
<td>DATA LD M+B</td>
</tr>
<tr>
<td>02FF 49</td>
<td>0030</td>
<td>LD L+C</td>
</tr>
<tr>
<td>0300 3B</td>
<td>0031</td>
<td>DEC SP</td>
</tr>
<tr>
<td>0301 3B</td>
<td>0032</td>
<td>DEC SP</td>
</tr>
<tr>
<td>0302 E3</td>
<td>0033</td>
<td>EX (SP)+HL</td>
</tr>
<tr>
<td>0303 FR</td>
<td>0034</td>
<td>EI</td>
</tr>
<tr>
<td>0304 ED4D</td>
<td>0035</td>
<td>RETI</td>
</tr>
<tr>
<td>0306 21CF05</td>
<td>0036</td>
<td>RET2: LD HL+RETNR CHANGE RETURN ADDRESS</td>
</tr>
<tr>
<td>0309 3B</td>
<td>0037</td>
<td>DEC SP</td>
</tr>
<tr>
<td>030A 3B</td>
<td>0038</td>
<td>DEC SP</td>
</tr>
<tr>
<td>030B E3</td>
<td>0039</td>
<td>EX (SP)+HL</td>
</tr>
<tr>
<td>030C FR</td>
<td>0040</td>
<td>EI</td>
</tr>
<tr>
<td>030D ED4D</td>
<td>0041</td>
<td>RETI</td>
</tr>
</tbody>
</table>

Errors 0
Table D2. (continued.)

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001 ;</td>
<td>;</td>
</tr>
<tr>
<td>0002 ;</td>
<td>;</td>
</tr>
<tr>
<td>0003 ; JUMP INSTRUCTION TO BEGINNING OF MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td>0004 ; ON RESET</td>
<td></td>
</tr>
<tr>
<td>0005 ;</td>
<td>;</td>
</tr>
<tr>
<td>0006 ;</td>
<td>;</td>
</tr>
<tr>
<td>(0400)</td>
<td>0007 MAIN EQU 400H</td>
</tr>
<tr>
<td>0008 ;</td>
<td>;</td>
</tr>
<tr>
<td>0000 C30004</td>
<td>0009 JP MAIN</td>
</tr>
</tbody>
</table>

Errors 0

A.
Errors 0

A.

Table D2. (continued.)
Table D2. (concluded.)

CROMEMCO CDOS Z80 ASSEMBLER version 02.15

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

Errors 0

A.
APPENDIX E
Z80 CPU INSTRUCTION SET*

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

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

SELECT PORT NUMBER:

P.A. 02H (WR)

2 3

4 5

6 7

P.A. 07H (WR)

3-INPUT NAND GATES

UI2, UI3 (74LS10)

309

D0

D7

P.A. 07H (RD)

P.A. 07H (RD)

P.A. 0Fh (RD)

S0-S2

S3-S5

S6-S8

P.A. 07H (RD)

DISPLAY SELECTION

P1

X2

D1

D2

P2

X2

P1

D1

D2

P2

X2

8 (D0-D7)
This Page Intentionally Left Blank
Figure F4

COMMUNICATIONS WITH MAIN COMPUTER

REQUEST A (LOW BYTE)
(NO CONNECTION)
DATA TRANS
STROBE

PORT CONTROL

REQUEST B (HIGH BYTE)

PA 0 (OUT 00)
PA 3 (OUT 03)

DRII-C INTERFACE MODULE

NEW DATA RDY

INTERRUPT

CPU UNIBUS

PDP II/34
COMPUTER

8 BIT PARALLEL OUTPUT LATCH
U38 (8212)

8 BIT PARALLEL OUTPUT LATCH
U39 (8212)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)

PORT ADDRESS 0EH (RD)
(PA0-PA3)

8

(D0-D7)
Figure F5

STEP GAIN SWITCH CODE MULTIPLEXERS AND READ BUFFER REGISTERS

INPUTS FROM PMT FUNCTION GENERATOR

3 TO 1 TIMES 12 SELECT
U51-U56 (74LS153)

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

PM0 PM7
PM8
PM11

U69, U70

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)

3 TO 1 SELECT
U58 (74LS153)

3 TO 1 TIMES 3 SELECT
U57,58 (74LS153)

READ BUFFERS
GR1 GR2
D4-2
D4-2

U60 THRU U65

INPUTS FROM PMT FUNCTION GENERATOR

GAIN SWITCH POSITION INPUTS
Y1-1
Y1-3

STEP CONTROL SIGNALS
S0-1
S0-3
S2-1
S2-3

START/STOP STEP MARKERS
LINE RECEIVERS U66 (75122)
REFERENCES


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.
End of Document