High-Speed Assembly Language
(80386/80387) Programming
for Laser Spectra Scan
Control and Data Acquisition
Providing Improved Resolution
Water Vapor Spectroscopy

Robert J. Allen
Allen Associates
Newport News, Virginia

Prepared by Allen Associates
for Vigyan Research Associates, Inc., for
NASA Langley Research Center
under Contract NAS1-17919

NASA
National Aeronautics
and Space Administration
Scientific and Technical
Information Division

1988
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>WATER VAPOR SPECTROSCOPY EXPERIMENTS</td>
<td>1</td>
</tr>
<tr>
<td>CHOICE OF ASSEMBLY LANGUAGE PROGRAMMING</td>
<td>2</td>
</tr>
<tr>
<td>DESCRIPTION OF PROGRAMS</td>
<td>3</td>
</tr>
<tr>
<td>FORMULATING THE PROGRAMS</td>
<td>5</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>6</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>6</td>
</tr>
</tbody>
</table>

LIST OF FIGURES

Figure

1. BLOCK DIAGRAM, WATER VAPOR SPECTROSCOPY EXPERIMENT ....................... 7
2. FLOW DIAGRAMS:
   A. OF THE BASIC PROGRAM .............................................. 8
   B. OF THE ASSEMBLY LANGUAGE PROGRAM .................................. 9
3. REPRESENTATIVE WATER (H2O) SPECTRUM CURVES:
   A. Cross Section vs. Wavenumber showing pressure shift in air for the line centered at 13914.9502 cm⁻¹ ............................................. 11
   B. Cross Section vs. Wavenumber showing pressure shift in air for the line centered at 13947.2608 cm⁻¹ ............................................. 12
4. SCOPE PHOTOGRAPHS OF 5-VOLT PULSES GENERATED FROM BINARY OUTPUTS BO-0 TO BO-3. PULSE WIDTHS AND VARIOUS PROGRAM TIMES ARE INCLUDED ............ 13
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>PROGRAM TIMES DETERMINED USING BINARY OUTPUT PULSES SHOWN IN FIGURE 4</td>
<td>15</td>
</tr>
<tr>
<td>II.</td>
<td>BASIC PROGRAM (ALLENB7R.BAS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A. LISTING OF PROGRAM</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>B. DISPLAY PRODUCED BY RUNNING PROGRAM</td>
<td>19</td>
</tr>
<tr>
<td>III.</td>
<td>ASSEMBLY LANGUAGE PROGRAMS:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A. ALLEN7R.LST (Microsoft Macro Assembler, Version 4.00)</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>B. SYMDEB ALLEN7R.EXE (Microsoft Symbolic Debug Utility, Version 4.00)</td>
<td>27</td>
</tr>
</tbody>
</table>
ABSTRACT

An assembly language program using the Intel 80386 CPU and 80387 math co-processor chips was written to increase the speed of data gathering and processing, and provide control of a scanning CW ring dye laser system. This laser system is used in high resolution (better than 0.001 cm⁻¹) water vapor spectroscopy experiments. Laser beam power is sensed at the input and output of white cells and the output of a Fabry-Perot. The assembly language subroutine is called from Basic, acquires the data and performs various calculations at rates greater than 150 times faster than could be performed by the higher level language. The width of output control pulses generated in assembly language are 3 to 4 microseconds as compared to 2 to 3.7 milliseconds for those generated in Basic (about 500 to 1000 times faster).

Included are a block diagram and brief description of the spectroscopy experiment, a flow diagram of the basic and assembly language programs, listing of the programs, scope photographs of the computer generated 5-volt pulses used for control and timing analysis, and representative water spectrum curves obtained using these programs.
HIGH-SPEED ASSEMBLY LANGUAGE (80396/80387) PROGRAMMING
FOR LASER SPECTRA SCAN CONTROL AND DATA ACQUISITION
PROVIDING IMPROVED RESOLUTION WATER VAPOR SPECTROSCOPY

By
Robert J. Allen

INTRODUCTION

This report describes an Intel 80386 and 80387 assembly language program written to increase the speed of data gathering and processing, and provide control of a scanning CW ring dye laser system. This laser system is used in high resolution Water Vapor Spectroscopy Experiments. The 80386 is a 32-bit CPU chip operating at 16 MHz in a Compaq 386 computer, and the 80387 is a high speed math coprocessor. The assembly language subroutine is called from BASICA.

The program was originally written for the Compaq Portable 286 computer using the 80286 and 80287 chips. It was later updated for use with the faster Compaq 386.

WATER VAPOR SPECTROSCOPY EXPERIMENTS

A block diagram of the spectroscopy experiment is shown in Fig. 1. An argon laser pumps a tunable CW ring dye laser. After receiving a Laser Scan Trigger, the Scanning Electronics will cause a very narrow line-width (0.0001 pm) laser beam to be outputted at wavelengths within pre-selected limits, for

# Figures 1 and 3 were provided by Dr. Benoist Grossmann, project scientists conducting the water vapor spectroscopy experiments.
example, 726.000 to 726.050 nm. This 50 pm scanning range can be selected anywhere between 725 nm (13,793.1 cm\(^{-1}\)) to 730 nm (13,698 cm\(^{-1}\)). A portion of the output beam is directed by beamsplitters to a Monochrometer for absolute wavelength calibration, a Fabry-Perot and photodiode P4 to provide the wavelength markers, and P3 as the Power Reference. The remaining beam is routed through White Cells #1 and #2 and sensed by photodiodes P1 and P2.

The outputs from the four photodiodes, P1-P4, are amplified and routed to four analog input channels multiplexed into a 10 KHz analog-to-digital converter with 12-bit resolution (A/D Board). The four multiplexed digital outputs are then processed as described later in this report. The A/D Board is part of an IBM Data Acquisition and Control Adapter card that fits inside the Compaq. This Adapter also contains two analog output channels; a 16-bit digital input port; a 16-bit digital output port; a 32-bit timer; and a 16-bit, externally-clocked, timer/counter. One bit of the digital output port is available for use as the Laser Scan Trigger.

**CHOICE OF ASSEMBLY LANGUAGE PROGRAMMING**

Since the four readings from photodiodes P1-P4 averaged N times are a single point on a rapidly changing plot of a single water vapor line, this data set should be obtained as rapidly as the hardware allows. More time, if needed, can be used to start the repeat (R) of subsequent data sets since this time is the separation between data points. Separation time, however,
should also be kept to a minimum since it is associated with the resolution of the curve.

As indicated in the general literature and by Rollins (1985), "assembly language is by far the fastest, most flexible, and most compact of all programming languages. It shows how and why higher-level languages operate. It gives access to features of a machine that are inaccessible with other languages." An assembly language matrix multiplication routine published by Startz (1985) is about 150 times faster than pre-8087 Basic.

 Fortunately, assembly language routines are easily combined with either interpreted or compiled Basic, as well as with programs written in other high-level languages. The main program illustrated in this report was written in Basic (BASICA.COM) calling the assembly language routine for acquiring N sets of the four photodiode (P1-P4) readings and performing required calculations.

DESCRIPTION OF PROGRAMS

The basic program (Table II) and assembly language program (Table III) were prepared as part of this task and provided to the Project Scientists. He incorporated them into his Basic program which included plotting features for use in the high resolution spectroscopy experiments.

Figure 2 contains a flow diagram of both programs. Basic is used first (prior to scanning) to calculate and display the 'mean' value of the background noise (W, X, Y & Z) from each of the four photodiodes (P1-P4). Scanning is initiated and the assembly language program called. The output from each of the
photodiodes are read and the corresponding background noise subtracted. The differences from P1, P2 and P4 are then normalized to the power reference difference P3. This process is repeated N times and the 'mean' of each of the three ratios \(((P1-X)/(P3-W)), (P2-Y)/(P3-W), \text{and } (P4-Z)/(P3-W)) \) calculated. The process then returns to Basic where the 'mean' of these three ratios are displayed. Each of the ratios constitute a single data point on each of three curves. The above process is then repeated R times obtaining new data points while the laser continues to scan.

BASICA line numbers are then displayed for the background sample size (M), data sample size (N) and number of repeats during laser scanning (R). The results, when coupled with the plotting program, are the curves shown in Figures 3A and B.

Five-volt pulses, BO-0 and BO-1, are generated in Basic prior to scanning and can be used to trigger a scope and to initiate scanning. Binary output pulses, BO-2 and BO-3, are generated in Assembly Language and were used to measure the time for different processes during program development such as the:

(1) greater-than 20 microsecond delay time required to allow transients to settle following each multiplexed operation, (2) total time used to read the four analog input from the photodiodes and subtract the backgrounds from each, and (3) time to calculate the mean of the ratios. Scope photographs of pulses BO-0 to BO-3, their pulse widths, and various times are shown in Figure 4 and Table I.
FORMULATING THE PROGRAMS

The assembly language program was written using WordStar, assembled using Microsoft Macro-Assembler version 4.00 (MASM ALLEN7R) and linked for high memory residence (LINK/H ALLEN7R). Debug was called (SYMDEB ALLEN7R.EXE) and Basic entered using N BASICA.COM. L and G. ALLEN7R.BAS was then loaded and a binary file created using DEF SEG = &H9FCA followed by BSAVE "ALLEN7R.BIN",0,&h178. Running ALLEN7R.BAS produced the display shown in Table IIB.
ACKNOWLEDGMENT

Publications by Lafore (1984) and Startz (1985) provided excellent references supporting the assembly language portions of this task. The author wishes to thank Dr. Jerry Tucker of NASA Langley Research Center for his helpful suggestion in connection with the assembly language programming, Dr. Benoist Grossmann for providing the information concerning the experiments and review of this report, and Patrick Ponsardin for his review and comments concerning this report. Both Benoist and Patrick are affiliated with Old Dominion University Research Foundation (on leave from Electricite de France).

REFERENCES


FIGURE 1. BLOCK DIAGRAM, WATER VAPOR SPECTROSCOPY EXPERIMENT.
FIGURE 2A. FLOW DIAGRAM OF THE BASIC PROGRAM.
FROM FIG. 2A

A

SET UP STACK FOR RETURN

SELECT BINARY OUTPUT DEVICE #2.
GENERATE 5 VOLT PULSE, BINARY OUTPUT 2 (BO-2) *

CLEAR MEMORY OFFSET LOCATIONS
210h THROUGH 2B7h

PREPARE TO TAKE 4N READINGS

HOLD NUMBER OF READINGS IN REGISTER CX

SELECT BINARY OUTPUT DEVICE #2.
GENERATE 5 VOLT PULSE BINARY OUTPUT 3 (BO-3) *

SELECT ANALOG INPUT DEVICE #9.
PREPARE TO TAKE ANALOG INPUT DATA
(SET BX = 0 FOR FIRST AD #)

CONNECT AD # DELAY GREATER THAN 20
MICROSECONDS TO ALLOW TRANSIENTS TO SETTLE.
START CONVERSION AND POLL FOR END-OF-
CONVERSION. READ AD # INPUT.

SUBTRACT CORRESPONDING BACKGROUND FROM AD# INPUT.
STORE DIFFERENCE IN MEMORY. INCREMENT BX.

IS BX > 37? NO

CALCULATE RATIOS: AD1-Z/AD1-W, AD2-Y/AD#-W, AND
AD3-Z/AD#-W. SUM THE THREE RATIOS. DECREMENT CX.

DOES CX = 0? NO

CALCULATE MEAN OF THE THREE RATIOS AND
CONVERT TO INTEGER (MULTIPLY BY 10,000)

RETURN TO BASIC * REMOVED FROM PROGRAM
USED TO PLOT FIGURES 3A AND 3B.

TO FIG. 2A

FIGURE 2B. FLOW DIAGRAM OF THE 80386/80387 ASSEMBLY
LANGUAGE PROGRAM.
FIGURE 3A. CROSS SECTION VS. WAVENUMBER SHOWING PRESSURE SHIFT IN AIR FOR THE LINE CENTERED AT 13914.9502 cm⁻¹.
**H2O Spectrum**

Pressure Shift In Air

10 torr H2O
10 torr H2O + 1 Atm Air
Fabry-Perot FSR = 0.100 cm⁻¹

**FIGURE 3B.** CROSS SECTION VS. WAVENUMBER SHOWING PRESSURE SHIFT IN AIR FOR THE LINE CENTERED AT 13947.2608 cm⁻¹.
FIGURE 4. SCOPE PHOTOGRAPHS OF 5-VOLT PULSES GENERATED FROM BINARY OUTPUTS BO-0 TO BO-3. PULSE WIDTHS AND VARIOUS PROGRAM TIMES ARE LISTED IN TABLE I.
(E) BO-1
2V/div  100 ms/div

(added program: 1395 GOTO 1060)

(F) BO-2
2V/div  1 ms/div

(G) BO-3
2V/div  1 ms/div

(added program: 1185 GOTO 1160)

ORIGINAL PAGE IS OF POOR QUALITY.

(H) BO-1 ADDED TO INTENSIFIED BO-2

(added program: 1185 GOTO 1060)
<table>
<thead>
<tr>
<th>BASICA LINE NUMBERS</th>
<th>ADDED PROGRAM STEP *</th>
<th>GENERATED BINARY OUTPUT PULSE</th>
<th>TIME FROM START OF BO-1 TO START OF BO-j</th>
</tr>
</thead>
<tbody>
<tr>
<td>80-140</td>
<td>1035 GOTO 80</td>
<td>BO-0, 5V 3.7 ms wide (FIG. 4A)</td>
<td>BO-0 to BO-0: 200 ms (M=1) 412 ms (M=10) 875 ms (M=30) 2500 ms (M=100)</td>
</tr>
<tr>
<td>1060-1100</td>
<td>1105 GOTO 1060</td>
<td>BO-1, 5V 2 ms wide (FIG. 4B)</td>
<td>BO-1 to BO-1: 3 ms</td>
</tr>
<tr>
<td>80-1100</td>
<td>DELETE 1040</td>
<td>BO-0 and BO-1 (FIG. 4C&amp;D)</td>
<td>BO-0 to BO-1: 200 ms (M=1) BO-1 to BO-0: 6.8 ms</td>
</tr>
<tr>
<td>1060-1390</td>
<td>1395 GOTO 1060</td>
<td>BO-1 (FIG. 4E)</td>
<td>BO-1 to BO-1: 770 ms</td>
</tr>
<tr>
<td>1160-1185</td>
<td>1185 GOTO 1160</td>
<td>BO-2 &amp; BO-3 3.5V into 10K</td>
<td>BO-2 to BO-2: 4.5 ms (N=1) 6.5 ms (N=3) 8.5 ms (N=5) 43.2 ms (N=40)</td>
</tr>
<tr>
<td></td>
<td>includes total</td>
<td></td>
<td>FOR N=5: BO-2 to BO-3 -- 0.31 ms BO-31 to BO-32 -- 1.0 ms BO-35 to BO-2 -- 4.2 ms</td>
</tr>
<tr>
<td></td>
<td>assembly language</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subroutine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1060-1185</td>
<td>1185 GOTO 1060</td>
<td>BO-1 ADDED TO INTENSIFIED</td>
<td>BO-1 to BO-2: -- 29.6 ms BO-2 to BO-1: -- 10.4 ms BO-1 to BO-1: -- 40 ms</td>
</tr>
</tbody>
</table>

*Added program to produce a repetitive pulse for timing analyses and scope photographic purposes.
TABLE IIA  BASIC PROGRAM ALLENB7R.BAS

10 CLS  '--- WATER VAPOR SPECTROSCOPY EXPERIMENT, CONTROL & DATA PROCESSING ---
20   'GENERATES BINARY OUTPUT PULSE BO 0, CALCULATES THE MEAN OF M SETS OF THE
30   BACKGROUND DATA FROM AD0-AD3, POKEs THE FOUR BACKGROUND VALUES & THE DATA
40   SAMPLE SIZE N INTO MEMORY, CONVERTS THE FOUR INTO A VOLTAGE, AND DISPLAYS THEIR
50   VALUES.
60   ' CALLS THE ASSEMBLY LANGUAGE PROGRAM. AFTER RETURNING TO BASIC, DIVIDES
70   THE THREE CALCULATED DATA RATIOS (corrected
80   for background noise) BY 10,000,
90   AND DISPLAYS THESE RATIOS.
100  ' THE ASSEMBLY LANGUAGE PROGRAM IS AGAIN CALLED AND THE PROCESS REPEATED
110  R TIMES. THE PROGRAM LISTING FOR THE BACKGROUND SAMPLE SIZE (M), DATA SAMPLE
120  SIZE (N) AND THE NUMBER OF REPEATS DURING LASER SCANNING (R) ARE DISPLAYED FOR
130  EASE OF CHANGE.
140   '********** GENERATE BINARY OUTPUT PULSE BO 0 **************
150 80 OUT 49890!,8  'Selects Reg. 12, Device 8
160 90 OUT 8930,0  'Zero volts from BO 0 (low byte)
170 100 OUT 8931,0  'Zero volts from BO 0 (high byte)
180 110 OUT 8930,1  '5 volts from BO 0 (low byte)
190 120 OUT 8931,0  ' (high byte) SCOPE TRIG +
200 130 OUT 8930,0  'Zero volts from BO 0 (low byte)
210 140 OUT 8931,0  ' (high byte) START MEASURING BACKGROUND
220 150 ' ********** TAKE THE MEAN OF BACKGROUND READINGS FROM AD0-AD3 **********
230 160 OUT 49890!,9  'BASIC ADDRESS OF DATA ACQ ADAPTER 0 = 2E2H (AD# INPUTS)
240 170 M = 30  'BACKGROUND SAMPLE SIZE, M
250 180 AL=0:AH=0:BL=0:BH=0:CL=0:DH=0
260 190 WL=0:WH=0:XL=0:YL=0:ZH=0:ZL=0
270 200 FOR J=1 TO M
280 210 OUT 738,0 '
290 220 OUT 738,0  'SELECT AD 0 (P1)
300 230 OUT 739,0
310 240 OUT 738,1
320 250 OUT 739,0
330 260 OUT 738,0
340 270 OUT 739,0
350 280 AL=INP (8930)  'READ AD 0
360 290 WL=WL+AL
370 300 AH=INP (8931)
380 310 WH=WH+AH  'SUM AD 0
390 320 OUT 738,0  'SELECT/READ/SUM AD 1 (P2)
400 330 OUT 739,1
410 340 OUT 738,1
420 350 OUT 739,1
430 360 OUT 738,0
440 370 OUT 739,1
450 380 BL=INP (8930)
460 390 XL=XL+BL
470 400 BH=INP (8931)
480 410 XH=XH+BH  'SELECT/READ/SUM AD 2 (P3)
490 420 OUT 738,0
500 430 OUT 739,2
510 440 OUT 738,1
520 450 OUT 739,2
530 460 OUT 738,0
540 470 OUT 739,2
550 480 CL=INP (8930)
560 490 YL=YL+CL
570 500 CH=INP (8931)
580 510 YH=YH+CH
520 OUT 738,0 'SELECT/READ/SUM AD 3 (P4)
530 OUT 739,3
540 OUT 738,1
550 OUT 739,3
560 OUT 738,0
570 OUT 739,3
580 DL=INP (8930)
590 ZL=ZL+DL
600 DH=INP (8931)
610 ZH=ZH+DH
620 NEXT J 'REPEAT SELECT/READ/SUM DATA SET AD 0 - AD 3
630 '  
640 WL=WL/M 'MEAN OF BACKGROUND READINGS FROM AD 0
650 WH=WH/M
660 XL=XL/M 'MEAN OF BACKGROUND READINGS FROM AD 1
670 XH=XH/M
680 YL=YL/M 'MEAN OF BACKGROUND READINGS FROM AD 2
690 YH=YH/M
700 ZL=ZL/M 'MEAN OF BACKGROUND READINGS FROM AD 3
710 ZH=ZH/M
720 '  
730 K = 5 'DATA SAMPLE SIZE, N
740 '  
750 ' ********** STORE MEAN BACKGROUND READINGS AND N IN MEMORY **********
760 DEF SEG = &H9FCA 'SEGMENT ADDRESS (page 426)
770 ADDR = &H200
780 POKE ADDR, WL 'SAVE AD 0 BACKGROUND IN MEMORY 200H
790 ADDR=ADDR+1
800 POKE ADDR, WH
810 ADDR=ADDR+1
820 POKE ADDR, XL 'SAVE AD 1 BACKGROUND IN MEMORY 202H
830 ADDR=ADDR+1
840 POKE ADDR, XH
850 ADDR=ADDR+1
860 POKE ADDR, YL 'SAVE AD 2 BACKGROUND IN MEMORY 204H
870 ADDR=ADDR+1
880 POKE ADDR, YH
890 ADDR=ADDR+1
900 POKE ADDR, ZL 'SAVE AD 3 BACKGROUND IN MEMORY 206H
910 ADDR=ADDR+1
920 POKE ADDR, ZH
930 ADDR = &H208
940 POKE ADDR, N 'SAVE DATA SAMPLE SIZE, N, IN MEMORY 208H
950 '  
960 ' ********** CONVERT TO VOLTAGE AND DISPLAY **********
970 W=.00244*(256*WH+WL)
980 X=.00244*(256*WH+XL)
990 Y=.00244*(256*WH+YL)
1000 Z=.00244*(256*WH+DL)
1010 PRINT "BACKGROUND", "W", "X", "Y", "Z"
1020 PRINT ,W, X, Y, Z
1030 PRINT
1040 STOP
1050 ' ********** GENERATE BINARY OUTPUT PULSE BO1 **********
1060 OUT 498901,8 'Selects Reg. 12, Device 8
1070 OUT 8930,2 '5 volts from BO 1 (low byte)
1080 OUT 8931,0 ' (high byte) 4-CHANNEL MARKER
1090 OUT 8930,0 'Zero volts from BO 1 (LOW BYTE)
1100 OUT 8931,0 'Zero volts from BO 1 (high byte)
1110 '######### PREPARE FOR ASSEMBLY LANGUAGE PROGRAM #########
1120 BLOAD "ALLEN7R.BIN",0 '(R. L. Lafore, page 428)
1130 R = 300 'NUMBER OF REPEATS DURING LASER SCANNING, R
1140 R = 3 'NUMBER OF REPEATS DURING LASER SCANNING, R
1150 FOR S=1 TO R
1160 CALL ALLEN7R
1170 '######### GO TO ASSEMBLY LANGUAGE PROGRAM #############
1180 '---------- RETURN FROM ASSEMBLY LANGUAGE PROGRAM ------
1190 '********* DISPLAY DATA CALCULATED IN ASSEMBLY LANGUAGE *********
1200 DEF SEG =&H9FCA
1210 ADDR = &H270
1220 PRINT "DATA OFFSET ADDRESS", "RATIO"
1230 FOR K=1 TO 3
1240 EL = PEEK(ADDR)
1250 EH = PEEK(ADDR+1)
1260 E=(256*EH+EL)
1270 PRINT " " HEX$(ADDR),
1280 PRINT USING "##.##";E
1290 F=E/10000
1300 PRINT ",,F
1310 ADDR = ADDR+8
1320 NEXT K
1330 NEXT S
1340 PRINT
1350 '********** DISPLAY SAMPLE SIZE OF CALCULATED DATA **********
1360 PRINT "170 M ="M " BACKGROUND SAMPLE SIZE, M
1370 N = PEEK(&H208)
1380 PRINT "730' N ="N " DATA SAMPLE SIZE, N
1390 PRINT "1140 R ="R " NUMBER OF REPEATS DURING LASER SCANNING, R
1400 END
1410 'SAVE"ALLEN7R 11/3/87
OK
**TABLE IIB  DISPLAY PRODUCED BY RUNNING ALLENB7R.BAS**

Break in 1040

<table>
<thead>
<tr>
<th>DATA OFFSET</th>
<th>ADDRESS</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>270</td>
<td>%21154.0</td>
<td>2.1154</td>
</tr>
<tr>
<td>278</td>
<td>%31532.0</td>
<td>3.1532</td>
</tr>
<tr>
<td>280</td>
<td>%42287.0</td>
<td>4.2287</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA OFFSET</th>
<th>ADDRESS</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>270</td>
<td>%21053.0</td>
<td>2.1053</td>
</tr>
<tr>
<td>278</td>
<td>%31361.0</td>
<td>3.1361</td>
</tr>
<tr>
<td>280</td>
<td>%42071.0</td>
<td>4.2071</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA OFFSET</th>
<th>ADDRESS</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>270</td>
<td>%21055.0</td>
<td>2.1055</td>
</tr>
<tr>
<td>278</td>
<td>%31418.0</td>
<td>3.1418</td>
</tr>
<tr>
<td>280</td>
<td>%42143.0</td>
<td>4.2143</td>
</tr>
</tbody>
</table>

170 M = 30  'BACKGROUND SAMPLE SIZE, M
730' N = 5  'DATA SAMPLE SIZE, N
1140 R = 3  'NUMBER OF REPEATS DURING LASER SCANNING, R

Ok
Generates a BINARY OUTPUT PULSE (B02) to indicate the START of taking data, followed by a B03 pulse to be used as a marker at the start of each set of ADC INPUTS.

Reads data from AD0 and subtracts W (poked into RAM previously from BASICA) and holds the DIFFERENCE, then reads data from AD1 and subtracts X and holds this DIFFERENCE, etc. for AD2-Y and AD3-Z. This is repeat N time where 2>N>100 which was previously poked from BASICA. A maximum of four memory locations (200H, 202H, 204H, & 206H) are required for the BACKGROUND data previously averaged in BASIC; and four memory locations (210H, 218H, 220H, & 228H) for subtracted data, three memory locations (230H, 238H, & 240H) for the RATIO data, one (248H) for the value of N, three (250H, 258H, & 260H) for the summation of RATIOS, and the same three (250H, 258H, & 260H) for the MEAN of the RATIOS all obtained in this ASSEMBLY LANGUAGE program. The values are then multiplied by 10,000d to eliminate the decimal point. The program then returns to basic.

BASICA PROGRAM -- ALLEN7R.BAS

;***********************************************************************

ST_SEG SEGMENT STACK ;DEFINE STACK SEGMENT
    DB 20 DUP ('STACK ')
ST_SEG ENDS

;***********************************************************************

PROGNAM SEGMENT ;DEFINE CODE SEGMENT

MAIN PROC FAR ;MAIN PART OF PROGRAM
    ASSUME CS:PROGNAM,DS:NOTHING
    START: ;START EXECUTION ADDR
;SET UP STACK FOR RETURN (p 439)
PUSH BP ;SAVE BP
PUSH DS ;SAVE OLD DATA SEGMENT
PUSH SI ;SAVE OLD SI
MOV AX,CS ;DATA SEGMENT SAME AS
MOV DS,AX ; CODE SEGMENT
SUB AX,AX ;ZERO AX
PUSH AX ;SAVE IT ON STACK

;SELECT/HOLD DATA ACQ/CTRL ADAPT--BINARY
MOV DX,OC2E2H ;REG 12,ADPTR 0,LO B
MOV AX,8 ;BINARY DEVICE 8
OUT DX,AL ; LOW BYTE OUTPUT
MOV DX,OC3E3H ;REG 12,ADPTR 0,HI B
MOV AX,0 ;BINARY DEVICE 8
OUT DX,AL ; HIGH BYTE OUTPUT

;Select Binary Output (Write), register 2
; Set BO 2 TTL output low
MOV DX,22E2H ;REG 2,BIN OUT,LO B
MOV AL,0 ;BO 2 = 0 (LO & HI)
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,22E3H ;REG 2,BIN OUT,HI B
OUT DX,AL ;OUTPUT HIGH BYTE

; Set BO 2 TTL output high
MOV DX,22E2H ;REG 2,BIN OUT,LO B
MOV AL,4 ;BO 2 = 1, LO B
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,22E3H ;REG 2,BIN OUT,HI B
MOV AL,0 ;HIGH BYTE OF BO 2=0
OUT DX,AL ;OUTPUT HIGH BYTE

; Set BO 2 TTL output low
MOV DX,22E2H ;REG 2,BIN OUT,LO B
MOV AL,0 ;BO 2 = 0 (LO & HI)
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,22E3H ;REG 2,BIN OUT,HI B
OUT DX,AL ;OUTPUT HIGH BYTE

;Fill memory offset locations
MOV BX,0210H ;OFFSET FILL ADDR.
MOV DI,BX ; INTO REG. DI
MOV CX,80H ;# LOCATIONS TO FILL
MOV AX,0000H ;PUT # INTO AX

FILL: MOV [DI],AX ;FILL MEMORY WITH #s
INC DI ;INCREASE ADDR. BY 1
LOOP FILL ;REPEAT FILLING

;Prepare to Take 4N readings
MOV BX,0208H ;N POKED FROM BASIC 1s
MOV CX,[BX] ;# OF READ. TO AVERAGE
MOV BX,0210H ;OFFSET DATA ADDR.
MOV DI,BX ; INTO REG. DI
MOV AL,4
NEWSET: PUSH CX ;# OF ADC CHANNELS
;SELECT/HOLD DATA ACQ/CTRL ADAPT--BINARY
MOV DX,OC2E2H ;REG 12, ADPTR 0, LO B
MOV AX,8 ;BINARY DEVICE 8
OUT DX,AL ; LOW BYTE OUTPUT
MOV DX,OC3E3H ;REG 12, ADPTR 0, HI B
MOV AX,0 ;BINARY DEVICE 8
OUT DX,AL ; HIGH BYTE OUTPUT

;Select Binary Output (Write), register 2
; Set BO 3 TTL output high (WAS LOW)
MOV DX,OC2E2H ;REG 12, ADPTR 0, LO B
MOV AL,8 ;BO 3 = 1, LO B
OUT DX,AL ; LOW BYTE OUTPUT
MOV DX,OC3E3H ;REG 12, ADPTR 0, HI B
MOV AL,0 ;HIGH BYTE OF BO 3=0
OUT DX,AL ; OUTPUT HIGH BYTE

; Set BO 3 TTL output low
MOV DX,OC2E2H ;REG 2, BIN OUT, LO B
MOV AL,0 ;BO 3 = 0 (LO & HI)
OUT DX,AL ; OUTPUT LOW BYTE
MOV DX,OC2E3H ;REG 2, BIN OUT, HI B
OUT DX,AL ;OUTPUT HIGH BYTE

;SELECT/HOLD DATA ACQ/CTRL ADAPT--ANALOG INPUT
MOV DX,OC2E2H ;REG 12, ADPTR 0, LO B
MOV AL,9 ;DEVICE 9 LOW BYTE
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,OC2E3H ;REG 12, ADPTR 0, HI B
MOV AL,0 ;DEVICE 9 HIGH BYTE
OUT DX,AL ;OUTPUT HIGH BYTE

;PREPARE TO TAKE ANALOG INPUT DATA
SUB BX,BX ;ZERO BL (AD# = AD0)

;Select Analog Input Control Reg 0 (AD#)
; Connect AD#. Hold conv (reset bit #0 to 0)
NEXTAD: MOV DX,02E2H ;REG 0, AD# INPUT, LO
MOV AL,0 ;HOLD CONVERT
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,02E3H ;REG 0, AD# INPUT, HI
MOV AL,BL ;CODE FOR AD# 0 thru 3
OUT DX,AL ;OUTPUT HIGH BYTE

; Delay >20 us (allow transients to settle)
MOV CX,0022H ;MEASURED DELAY: 26us

TRANS: LOOP TRANS ;EXECUTE DELAY #1

; Start Conversion (set bit #0 to 1)
MOV DX,02E2H ;REG 0, AD# INPUT, LO
MOV AL,1 ;START CONVERT, LOW B
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,02E3H ;REG 0, AD# INPUT, HI
MOV AL,BL ;CODE FOR AD# 0 thru 3
OUT DX,AL ;OUTPUT HIGH BYTE

; Wait for BUSY STATE bit 0 = 0 (Polling)
MOV DX,02E2H ;REG 0, LOW BYTE
ANL DX,0001H ;MASK ALL BUT BIT 0
JNZ CONV ;REPEAT IF BIT 0 NOT 0

; Enable Reading of AD# Channel
MOV DX,02E2H ;REG 0, AD# INPUT, LO
MOV AX,0 ;ENABLE READING, LO B
OUT DX,AL ;OUTPUT LOW BYTE
MOV DX,02E3H ;REG O, A/D 0 INPUT, HI
MOV AL,BL ;CODE FOR AD# 0 thru 3
OUT DX,AL ;OUTPUT HIGH BYTE

; Read AD# Input (register 2—see p71)
MOV DX,22E2H ;REG 2, AD# INPUT, LO
IN AX,DX ;INPUT READING

; Subtract background from A/D inputs
PUSH DI ;SAVE DATA ADDR.
MOV DX,0200H ;BACKGROUND DATA
ADD DX,BX ;ADDRESS
ADD DX,BX ;+ 2 TIMES
MOV DI,DX ;BL (AD#)
MOV DX,[DI] ;GET BACKGROUND
SUB AX,DX ;DIFFERENCE IN AX

; Store Difference in memory
POP DI ;DIFFERENCE ADDRESS
MOV [DI],AX ;DIFFERENCE IN MEMORY
ADD DI,8 ;INCREMENT ADDR BY 8

; Obtain another AD# (channel # 1, 2 or 3)
INC BL ;INCREASE AD# BY 1
CMP BL,4 ;IS BL > 3 i.e. (= 4)
JNZ NEXTAD ;NO? THEN REPEAT

; Take three ratios:
; AD1-X/AD0-W, AD2-Y/AD0-W, AD3-Z/AD0-W
MOV BX,0210H ;ADDR OF DIVISOR AD0-W
MOV AX,BX ;COPY INTO AX
ADD AX,8 ;ADDRESS OF DIVIDEND
MOV DI,AX ;AD1-X IN DI
MOV CX,BX ;ADDR OF AD1-X/AD0-W
ADD CX,20H ;OFFSET FROM DIVISOR
```assembly
MOV SI,CX ; BY 20H (IN SI)  
MOV CX,3 ; 3 RATIOS  
NXRAT: FILD DWORD PTR [DI] ; LD DIVIDEND-80287  
FIDIV DWORD PTR [BX] ; DIVIDE INTEGER  
FSTP DWORD PTR [SI] ; STORE REAL & POP  
ADD DI,8 ; NEXT AD# DIFFERENCE  
ADD SI,8 ; NEXT AD# RATIO ADDR  
LOOP NXRAT ; CALCULATE NEXT RATIO  

; Sum the three ratios  
ADD BX,20H ; ADDR OF AD1-X/AD0-W  
MOV DI,BX ; IN DI  
MOV AX,BX ; ADDR SUM AD1-X/AD0-W  
ADD AX,20H ; 20H HIGHER ADDR  
MOV SI,AX ; IN SI  
MOV CX,3 ; 3 SUMS  
NXSUM: FLD DWORD PTR [SI] ; LD REAL SUM  
FADD DWORD PTR [DI] ; ADD REAL  
FSTP DWORD PTR [SI] ; STORE REAL  
ADD SI,8 ; ADDR OF NEXT RATIO  
ADD SI,8 ; ADDR OF NEXT SUM  
LOOP NXSUM ; CALCULATE NEXT SUM  

; Prepare to obtain another set of 4 AD# read.  
MOV BX,0210H ; OFFSET DATA ADDR.  
MOV DI,BX ; INTO PTR DI  
MOV BX,0000H ; SELECT AD# = AD0  
POP CX ; # READINGS REMAINING  
DEC CX ; DECREASE BY 1  
JCCXZ MEAN ; END IF CX = 0  
JMP NEWSET ; OTHERWISE LOOP  

; Calculate mean (divide each sum by N)  
MEAN: MOV CX,3 ; 3 GROUPS OF SUMS  
MOV BX,00208H ; ADDRESS  
MOV DI,BX ; PTR FOR N  
MOV BX,0250H ; OFFSET ADDR.  
MOV SI,BX ; INTO PTR SI  
NXMEAN: FILD DWORD PTR [SI] ; LOAD INTO 80287  
FIDIV DWORD PTR [DI] ; DIVIDE BY N  
FSTP DWORD PTR [SI] ; STORE & POP MEAN  
ADD SI,8 ; NEXT OFFSET SUM ADDR  
LOOP NXMEAN ; REPEAT FOR NEXT MEAN  

; Multiply by 10000d and change to integer  
MOV BX,0268H ; ADDRESS OF MULTIPLIER  
MOV AX,2710H ; MOV+ 10000d MULTIPL.  
MOV [BX],AX ; INTO ADDR. 0268H  
MOV DX,0250H ; OFFSET MEAN ADDR  
MOV DI,DX ; POINTED TO BY DI  
MOV DX,0270H ; ADDRESS OF ANSWER  
MOV SI,DX ; POINTED BY SI  
MOV CX,3 ; REPEAT FOR 3 MEANS  
```
CONINT: FLD DWORD PTR [DI]; LOAD MEAN IN 80287
FIMUL DWORD PTR [BX]; MULTIPLY X 10000D
FISTP DWORD PTR [SI]; STORE & POP ANS
ADD DI,8 ; POINT NEXT MEAN
ADD SI,8 ; POINT NEXT ANSWER
LOOP CONINT ; NEXT CONVERT2INTEGER

; Stop After 3 ANSWERS in memory--short integer
POP AX
POP SI
POP DS
POP BP
NOP
NOP
RET ; RTN BASIC

MAIN ENDP ; END OF MAIN PART OF PROGRAM
DB 400D DUP(?) ; RESERVE DATA AREA

;---------------------------------------------
; PROGNAM ENDS
;---------------------------------------------

END START ;END ASSEMBLY
Segments and Groups:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Align</th>
<th>Combine</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGNAM</td>
<td>0308</td>
<td>PARA</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>ST_SEG</td>
<td>00A0</td>
<td>PARA</td>
<td>STACK</td>
<td></td>
</tr>
</tbody>
</table>

Symbols:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Attr</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONINT</td>
<td>L NEAR</td>
<td>015E</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>CONV</td>
<td>L NEAR</td>
<td>00A8</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>FILL</td>
<td>L NEAR</td>
<td>0044</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>MAIN</td>
<td>F PROC</td>
<td>0000</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>MEAN</td>
<td>L NEAR</td>
<td>012E</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>NEWSLET</td>
<td>L NEAR</td>
<td>0055</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>NEXTAD</td>
<td>L NEAR</td>
<td>0088</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>NXMEAN</td>
<td>L NEAR</td>
<td>013B</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>NXRAT</td>
<td>L NEAR</td>
<td>00E0</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>NXSUM</td>
<td>L NEAR</td>
<td>010E</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>START</td>
<td>L NEAR</td>
<td>0000</td>
<td>PROGNAM</td>
</tr>
<tr>
<td>TRANS</td>
<td>L NEAR</td>
<td>0097</td>
<td>PROGNAM</td>
</tr>
</tbody>
</table>

288 Source Lines
288 Total Lines
35 Symbols

49350 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\BG>
<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>AX</th>
<th>BX</th>
<th>CX</th>
<th>DX</th>
<th>SP</th>
<th>BP</th>
<th>SI</th>
<th>DI</th>
<th>DS</th>
<th>ES</th>
<th>SS</th>
<th>CS</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH BP</td>
</tr>
<tr>
<td>0000</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH BP</td>
</tr>
<tr>
<td>0001</td>
<td>1E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,CS</td>
</tr>
<tr>
<td>0002</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH SI</td>
</tr>
<tr>
<td>0003</td>
<td>BCC8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DS,AX</td>
</tr>
<tr>
<td>0005</td>
<td>BED8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DS,AX</td>
</tr>
<tr>
<td>0007</td>
<td>2BC0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUB AX,AX</td>
</tr>
<tr>
<td>0009</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH AX</td>
</tr>
<tr>
<td>000A</td>
<td>BAE2C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,C2E2</td>
</tr>
<tr>
<td>000D</td>
<td>B80800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0008</td>
</tr>
<tr>
<td>0010</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>0011</td>
<td>BAE3C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,C3E3</td>
</tr>
<tr>
<td>0014</td>
<td>B80000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0000</td>
</tr>
<tr>
<td>0017</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>0018</td>
<td>BAE222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,22E2</td>
</tr>
<tr>
<td>001B</td>
<td>B80000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0000</td>
</tr>
<tr>
<td>001E</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>001F</td>
<td>BAE322</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,22E3</td>
</tr>
<tr>
<td>0022</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>0023</td>
<td>BAE222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,22E2</td>
</tr>
<tr>
<td>0026</td>
<td>B004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AL,04</td>
</tr>
<tr>
<td>0028</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>0029</td>
<td>BAE322</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,22E3</td>
</tr>
<tr>
<td>002C</td>
<td>B000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AL,00</td>
</tr>
<tr>
<td>002E</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>002F</td>
<td>BAE222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,22E2</td>
</tr>
<tr>
<td>0032</td>
<td>B000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AL,00</td>
</tr>
<tr>
<td>0034</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>0035</td>
<td>BAE322</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,22E3</td>
</tr>
<tr>
<td>0038</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>0039</td>
<td>BB1002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV BX,0210</td>
</tr>
<tr>
<td>003C</td>
<td>8BF8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>003E</td>
<td>B90000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV CX,0080</td>
</tr>
<tr>
<td>0041</td>
<td>B80000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0000</td>
</tr>
<tr>
<td>0044</td>
<td>8905</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>0046</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV CX</td>
</tr>
<tr>
<td>0047</td>
<td>E2FB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>0049</td>
<td>BB0802</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0008</td>
</tr>
<tr>
<td>004C</td>
<td>8B0F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>004E</td>
<td>BB1002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0008</td>
</tr>
<tr>
<td>0051</td>
<td>8BF8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>0053</td>
<td>B004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AL,04</td>
</tr>
<tr>
<td>0055</td>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH CX</td>
</tr>
<tr>
<td>0056</td>
<td>BAE2C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>0059</td>
<td>B80800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0008</td>
</tr>
<tr>
<td>005C</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>005D</td>
<td>BAE3C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV AX,0000</td>
</tr>
<tr>
<td>0060</td>
<td>B80000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOV DX,AL</td>
</tr>
<tr>
<td>0063</td>
<td>EE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUT DX,AL</td>
</tr>
<tr>
<td>Address</td>
<td>Instructions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0064</td>
<td>MOV DX, 22E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0067</td>
<td>MOV AL, 08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0069</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:006A</td>
<td>MOV DX, 22E3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:006D</td>
<td>MOV AL, 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:006F</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0070</td>
<td>MOV DX, 22E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0073</td>
<td>MOV AL, 09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0075</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0076</td>
<td>MOV DX, C2E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0079</td>
<td>MOV AL, 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:007A</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:007D</td>
<td>MOV DX, 02E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:007F</td>
<td>MOV AL, 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0080</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0083</td>
<td>MOV DX, 02E3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0085</td>
<td>MOV AL, BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0086</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0088</td>
<td>MOV CX, 0022</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:008B</td>
<td>LOOP 0097</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:008D</td>
<td>MOV DX, 02E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:008E</td>
<td>MOV AL, 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0091</td>
<td>MOV DX, 02E3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0093</td>
<td>MOV AL, BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0094</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0097</td>
<td>MOV DX, 02E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:0099</td>
<td>MOV AL, 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:009C</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:009E</td>
<td>MOV DX, 02E3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:009F</td>
<td>MOV AL, BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00A2</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00A4</td>
<td>MOV DX, 02E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00A5</td>
<td>IN AX, DX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00A8</td>
<td>AND AX, 0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00A9</td>
<td>JNZ 00A8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00AC</td>
<td>MOV DX, 02E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00AE</td>
<td>MOV AX, 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00B1</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00B4</td>
<td>MOV DX, 02E3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00B5</td>
<td>MOV AL, BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00B8</td>
<td>OUT DX, AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00BA</td>
<td>MOV DX, 22E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00BB</td>
<td>IN AX, DX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00BE</td>
<td>PUSH DI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00BF</td>
<td>MOV DX, 0200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00C0</td>
<td>ADD DX, BX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00C3</td>
<td>ADD DX, BX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00C5</td>
<td>MOV DI, DX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00CE</td>
<td>MOV DX, [DI]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00C7</td>
<td>SUB AX, DX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00C9</td>
<td>POP DI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00CB</td>
<td>MOV [DI], AX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00CD</td>
<td>ADD DI, 08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00CE</td>
<td>INC BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00D3</td>
<td>CMP BL, 04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9FCA:00D5</td>
<td>JNZ 0088</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9FCA:0151 BA5002  MOV DX, 0250
9FCA:0154 8BFA  MOV DI, DX
9FCA:0156 BA7002  MOV DX, 0270
9FCA:0159 8BF2  MOV SI, DX
9FCA:015B B90300  MOV CX, 0003
9FCA:015E 9B  WAIT
9FCA:015F D905  FLD DWord Ptr [DI]
9FCA:0161 9B  WAIT
9FCA:0162 DA0F  FIMUL DWord Ptr [BX]
9FCA:0164 9B  WAIT
9FCA:0165 DB1C  FISTP DWord Ptr [SI]
9FCA:0167 83C708  ADD DI, +08
9FCA:016A 83C608  ADD SI, +08
9FCA:016D E2EF  LOOP 015E
9FCA:016F 58  POP AX
9FCA:0170 5E  POP SI
9FCA:0171 1F  POP DS
9FCA:0172 5D  POP BP
9FCA:0173 90  NOP
9FCA:0174 90  NOP
9FCA:0175 CB  RETF
9FCA:0176 90  NOP
9FCA:0177 90  NOP
High-Speed Assembly Language (80386/80387) Programming for Laser Spectra Scan Control and Data Acquisition Providing Improved Resolution Water Vapor Spectroscopy

Robert J. Allen

Vigyan Research Associates, Inc.
28 Research Drive
Hampton, VA 23666

National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

Final Report in support of the LASE Project to redefine the water vapor lines with improved accuracy in the 725-730 nm region.


An assembly language program using the Intel 80386 CPU and 80387 math coprocessor chips was written to increase the speed of data gathering and processing, and provide control of a scanning CW ring dye laser system. This laser system is used in high resolution (better than 0.001 cm⁻¹) water vapor spectroscopy experiments. Laser beam power is sensed at the input and output of white cells and the output of a Fabry-Perot. The assembly language subroutine is called from Basic, acquires the data and performs various calculations at rates greater than 150 faster than could be performed by the higher level language. The width of output control pulses generated in assembly language are 3 to 4 microseconds as compared to 2 to 3.7 milliseconds for those generated in Basic (about 500 to 1000 times faster).

Included are a block diagram and brief description of the spectroscopy experiment, a flow diagram of the basic and assembly language programs, listing of the programs, scope photographs of the computer generated 5-volt pulses used for control and timing analysis, and representative water spectrum curves obtained using these programs.

Unclassified - Unlimited

Subject Category 60

Unclassified

Unclassified

38

A03