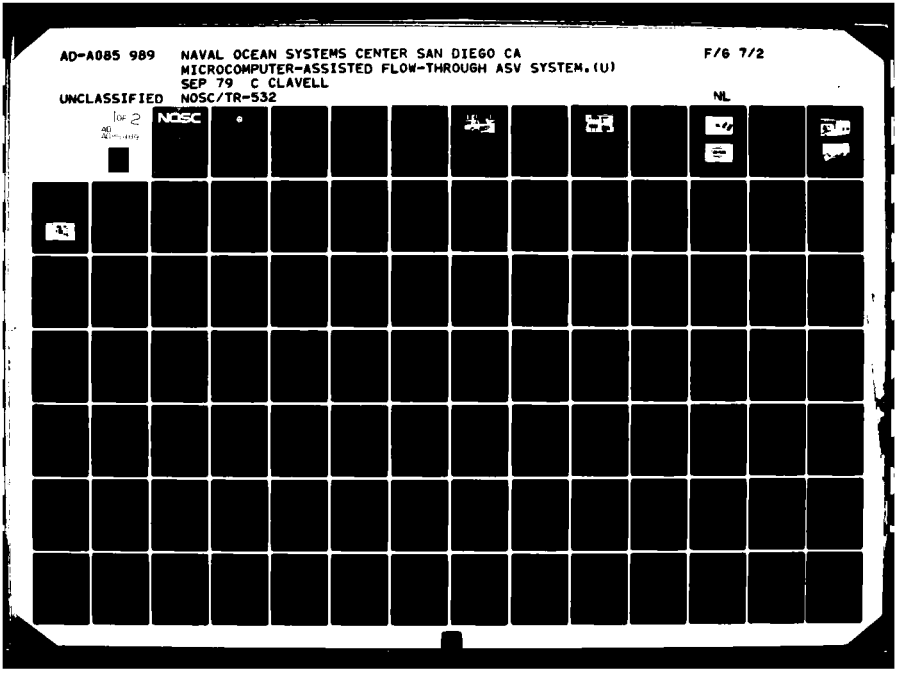
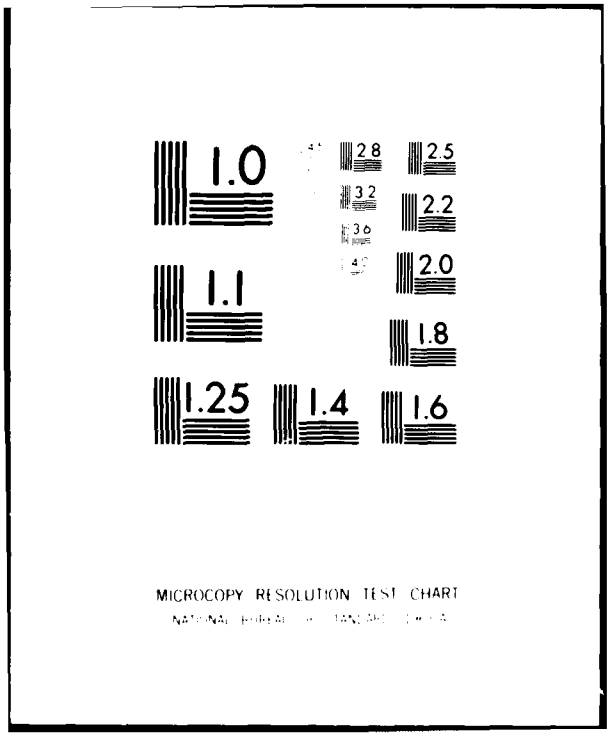
AD-A085 989 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 7/2
MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM.(U)
SEP 79 C CLAVELL
UNCLASSIFIED NOSC/TR-532 NL




1.0
2.8
2.5
3.2
2.2
3.6
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

LEVEL II

12

NOSC

NOSC TR 532

NOSC TR 532

AD A085989

Technical Report 532

MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM

C Clavell, Jr

September 1979

Final Report

Prepared for
Naval Material Command

DTIC
ELECTE
JUN 27 1980
S
C

Approved for public release;
distribution unlimited.

DDC FILE COPY

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

80 6 26 043

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

The work reported herein was sponsored by the Naval Material Command (63765N, F57572, ZF57572004, 513-ME02) and conducted over the period October 1976 to September 1979.

Released by
S Yamamoto, Head
Marine Sciences Division

Under authority of
HO Porter, Head
Biosciences Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NOSC Technical Report 532 (TR 532)

2. GOVT ACCESSION NO.: AD-A085989

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM

5. TYPE OF REPORT & PERIOD COVERED: Final Report, Oct[illegible]976-Sept[illegible]979

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): C. Clavell, Jr.

8. CONTRACT OR GRANT NUMBER(s): F57572

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Naval Ocean Systems Center
San Diego, California 92152

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
63765N, F57572, ZF57572004
513-ME02

11. CONTROLLING OFFICE NAME AND ADDRESS:
Naval Material Command
Technology Div., (NMAT-08T245)
Washington, D.C. 20360

12. REPORT DATE: Sept[illegible]979

13. NUMBER OF PAGES: 182

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 176

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

14 NOSC/TR-532

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A microcomputer has been adapted for use with a flow-through anodic stripping voltammetry (ASV) instrument developed by the Naval Ocean Systems Center. The ASV instrument was originally developed to provide near-real-time analysis of trace metals in seawater. Four years' experience with this system in the coastal waters of Florida, Peru, and California showed that many ancillary instruments, such as pH meters, specific ion meters, and fluorometers, could be added to better define sharply varying ocean environments. To coordinate the operations of these instruments and to deal with the very large amount of raw data that would be generated,

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

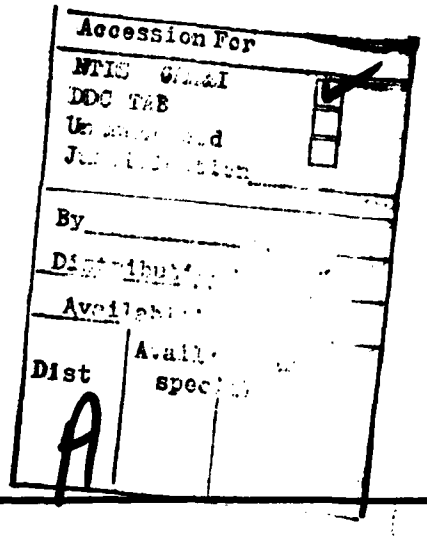
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393159

20. Abstract (Continued)

a Motorola 6800 microcomputer was added. Two major system software routines were developed for this application. The first controls ASV instrumentation, while the second is a monitor program that enables the system to function as a "stand alone" microcomputer. Thus equipped, the system should greatly facilitate trace metal investigations in the field by increasing the amount and accuracy of the data obtained while decreasing routine manual operations. Moreover, with substitution of a simulator for the wet-chemistry electromechanical unit, the system can serve as a tool for developing new software and hardware without requiring an investment in new, unproven equipment. The system can be expanded or modified within constraints of size, weight, and available memory.



SUMMARY

A second-generation, microcomputer-controlled flow-through anodic stripping voltammetry (ASV) instrument has been developed by the Naval Ocean Systems Center (NOSC). The first-generation, non-computerized instrument was originally developed to provide near-real-time analysis of trace metals in seawater. Four years' experience with this system in the coastal waters of Florida, Peru, and California indicated the need for added instrumental flexibility to permit changes in methodologies and to deal with the very large amount of raw data generated. To achieve these goals, a new system designed around a Motorola 6800 microcomputer was developed. Two major system software routines were written for this application: an instrument control program and a computing routine. The latter performs the necessary calculations to yield actual concentration values for each metal studied. Thus equipped, the system has improved the amount and accuracy of the data obtained while decreasing routine manual operations. Moreover, with the substitution of a simulator for the wet-chemistry electromechanical unit, the system can serve as a tool for developing new software and hardware without requiring investment in new, unproven equipment. The system can be expanded or modified within the constraints of size, weight, and available memory.

BACKGROUND

ASV is a form of polarography that has produced new interest in this field in recent years (Refs. 1-3). It is a technique that allows relatively fast measurements, to be made for various metals in different solutions without the need for elaborate preparations or preconcentration procedures. The technique's sensitivity is outstanding, having approached the parts-per-trillion level for several metals.

From the standpoint of field measurements, the main metals of interest are copper, lead, cadmium, and zinc. Using present equipment and techniques, it is possible to make simultaneous measurements of any three of these four metals directly in seawater with no pretreatment. It is this feature which has made ASV such an attractive tool for research and routine monitoring operations.

A generalized ASV instrument consists of (1) a potentiostat that controls the potential on the working electrode during electrolysis; (2) a cell arrangement to hold the sample and electrodes; and (3) some type of data display device, generally an X-Y recorder. Figure 1 illustrates a typical commercial system. Although there are several types of materials commonly used for the working electrode, this discussion will be limited to those composed of solid, glassy carbon. This electrode is most frequently rod-shaped, and either a short section of the rod or simply the face of one end is used as the active electrode surface. For either case, the surface to be use must be highly polished.

A generalized procedure for an ASV measurement of Cu, Pb and Cd begins with the application of a negative potential (vs SCE reference electrode) to the working electrode. The working electrode is in contact with the sample solution, to which a dilute mercury solution has been added. The applied potential causes a thin mercury metal film to be plated onto the electrode surface, and subsequently the metal ions in the sample are reduced and plated into the mercury film for several minutes. It is this electrolytic concentration which gives the technique its great sensitivity. After an appropriate time interval, the electrode is "scanned" by incrementing the potential in the positive direction. As the

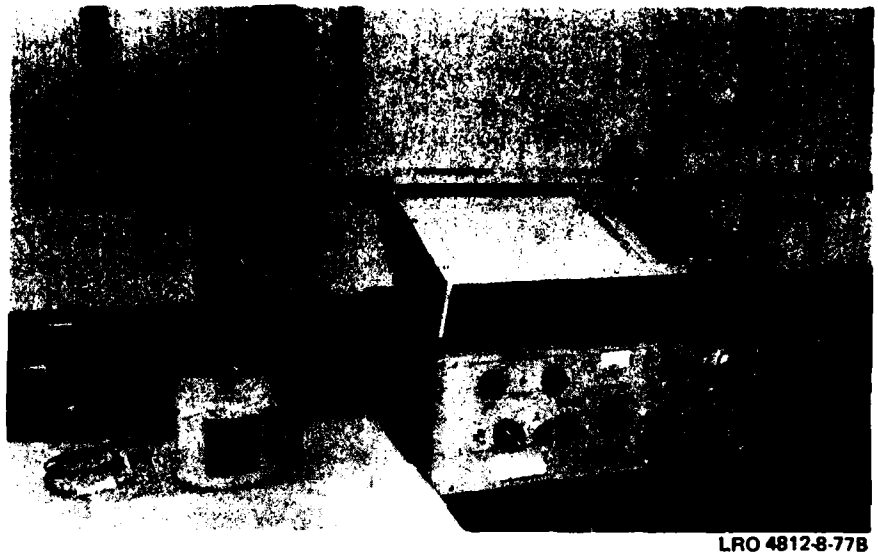


Figure 1. Commercially available system for ASV.

electrode voltage becomes less negative, the individual metals oxidize out of the mercury film, causing a current to flow. The current can be detected and displayed as a peak on the strip chart recorder (Fig. 2). Since each metal oxidizes out of the film at its own characteristic potential, it is possible to effect excellent separations in the multicomponent samples, with quantification being made by the technique of standard additions (Ref. 4).

There are many variations in both the methodology and the equipment currently used for ASV. Recently, new equipment incorporating microprocessors to control instrument parameters and aid in the display of results has come on the market. The Princeton Applied Research Model 374-1 is one example. Other instruments utilizing hard-wired programming designed for system flexibility have also been introduced recently, an example being Environmental Science Association's Model 3040. However, there is one trait common to all systems that are presently available commercially. They are basically laboratory instruments geared to measuring individual samples.

For the researcher interested in real-time events (biological phenomena, pollution surveys, etc.), such equipment is of little value, since the need to collect and process individual samples inherently precludes real-time analysis. An additional problem, which becomes extremely significant at the parts-per-billion (ppb) level, is sample contamination resulting from human intervention in the sample acquisition and during secondary handling at the man-instrument interface.

To overcome these problems, a flow-through ASV instrument system was developed at NOSC that provides near-real-time capability and requires no operator intervention in sampling (Ref. 6). As Fig. 3 shows, the system components are housed in four containers designed for field use. The potentiostat, the hard-wired programming unit, and the strip chart recorder are packaged in individual boxes, while the fourth box contains the chemical, electromechanical cell. The system can run unattended for periods of up to 48 hours, but suffers from lack of hardware flexibility. Thus, modification or expansion of the system is

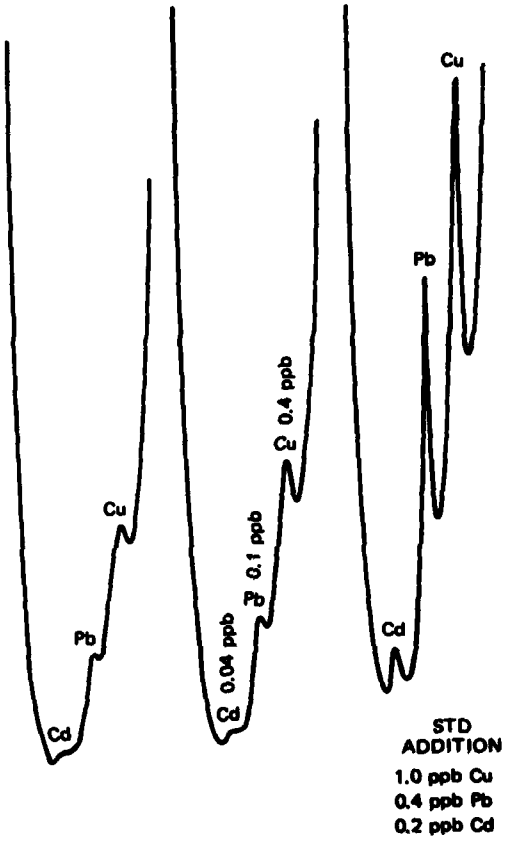


Figure 2. Typical traces from ASV.

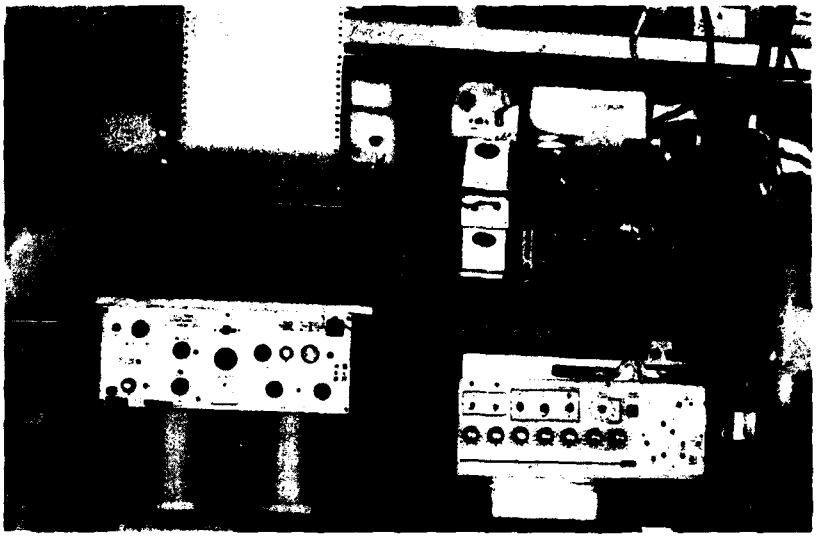


Figure 3. Automated real-time chemical analysis equipment.

very difficult. This particular instrument has been in service for approximately 4 years and was used in the coastal waters of Florida, Peru, and California.

The type of information provided by the flow-through instrument may be seen in Fig. 4. This figure shows the results of a 5-day, continuous operation conducted to measure zinc levels at a stationary point in San Diego Bay. The lower curves, consisting of 358 individual measurements, represent the raw data of relative peak current vs time; the upper trace is the processed data, giving the actual Zn concentrations in ppb vs time. It can be seen from these data that tidal fluctuations have a profound influence on the zinc concentration at any given moment and consequently could lead an investigator to draw erroneous conclusions about the zinc content of the bay if sampling had occurred at more widely spaced intervals.

In principle, this instrument operates like the general ASV instrument described earlier. Functionally, however, the instrument is unique and represents a significant advancement in the state of the art of trace-metal investigations in the field.

The major innovations consisted of the development of the wax-impregnated tubular graphite electrode (Ref. 5) (which has recently been supplanted by the flow-through carbon disk electrode) and the multielement tubular reference electrode (Ref. 6). Both have been patented by NOSC. The wax-impregnated tubular graphite electrode consists simply of a 1/4-in. high-purity graphite rod with a 1/8-in. hole drilled through the center. Before use, the graphite tube is vacuum-impregnated with paraffin, and the inner bore is sanded and polished to a high gloss (Fig. 5). A newly developed sensor, the carbon disk electrode (Fig. 5), consists of a Teflon holder that can accommodate two glassy carbon or LTI (low-temperature isotropic) carbon disks. Each disk is approximately 5/8 in. in diameter and 1/16 in. thick. As with the tubular graphite electrode, electrical contact with the disks is made by a Pt wire lead pressed against the surface. The multielement reference electrode (Fig. 6) incorporates an Ag/AgCl element and a platinum element within the same tubular

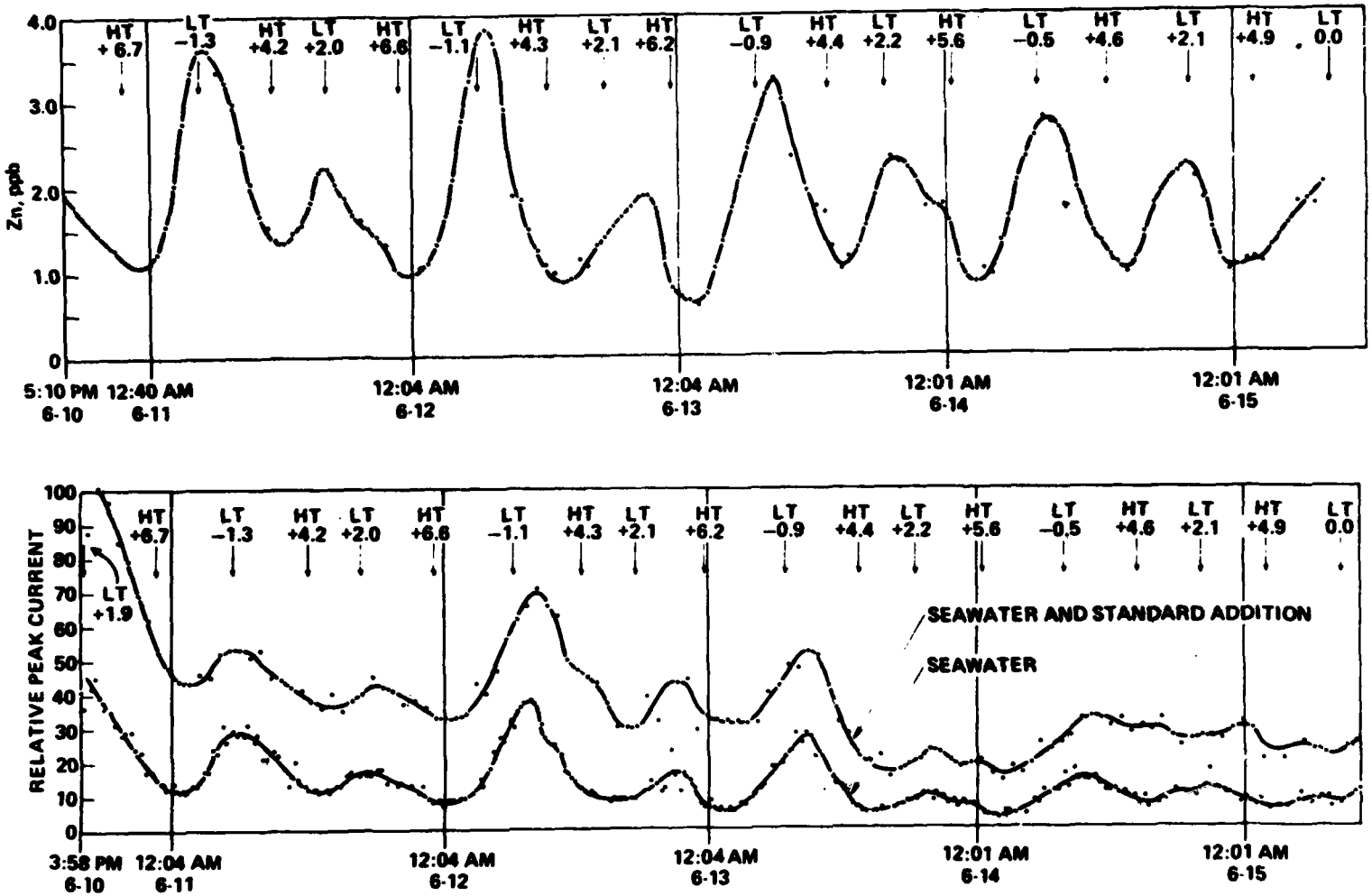
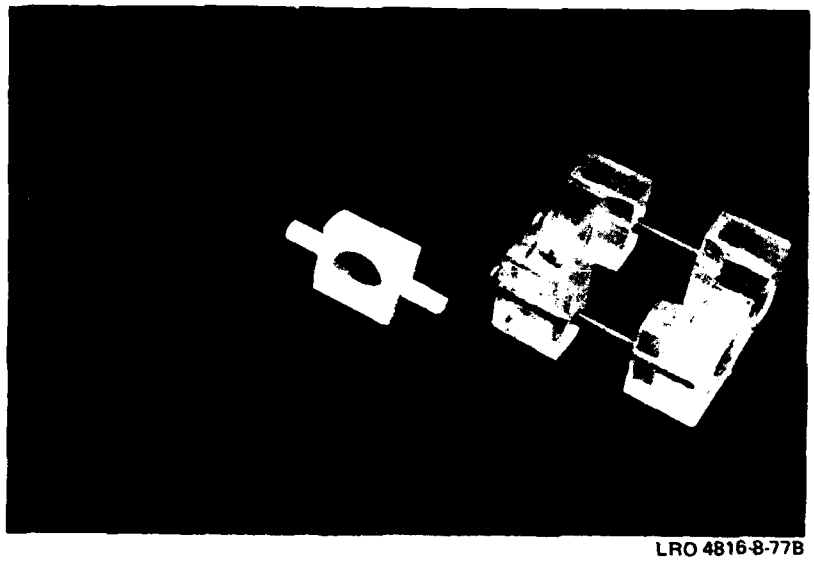


Figure 4. Zinc tracings made using flow-through instrument. Upper figure: Tidal fluctuations of Zn concentrations in San Diego Bay during 10–15 June 1975. Lower figure: Peak currents for Zn in seawater and seawater plus standard addition.



LRO 4816-8-77B

Figure 5. Working electrodes for flow-through instrument.

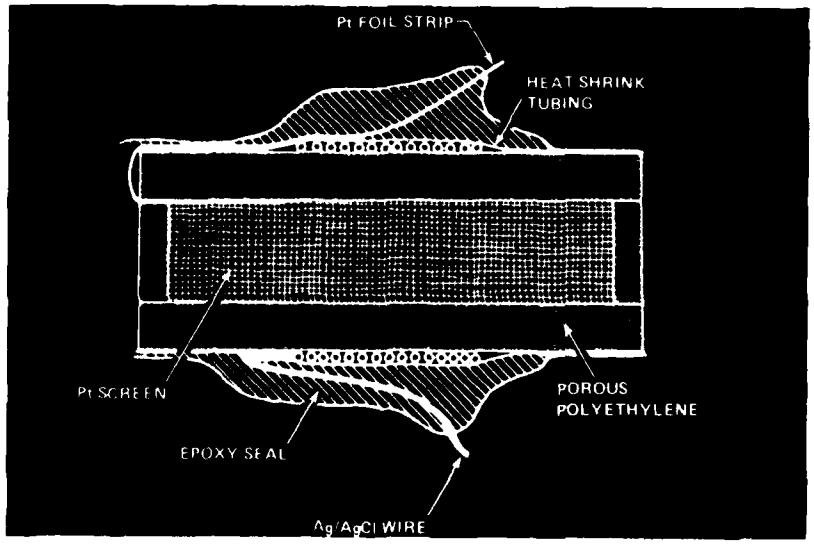


Figure 6. Reference electrode for flow-through instrument.

electrode. Isolating the Ag/AgCl element from the sample stream with a salt bridge has eliminated the problem of Ag^+ ion contamination.

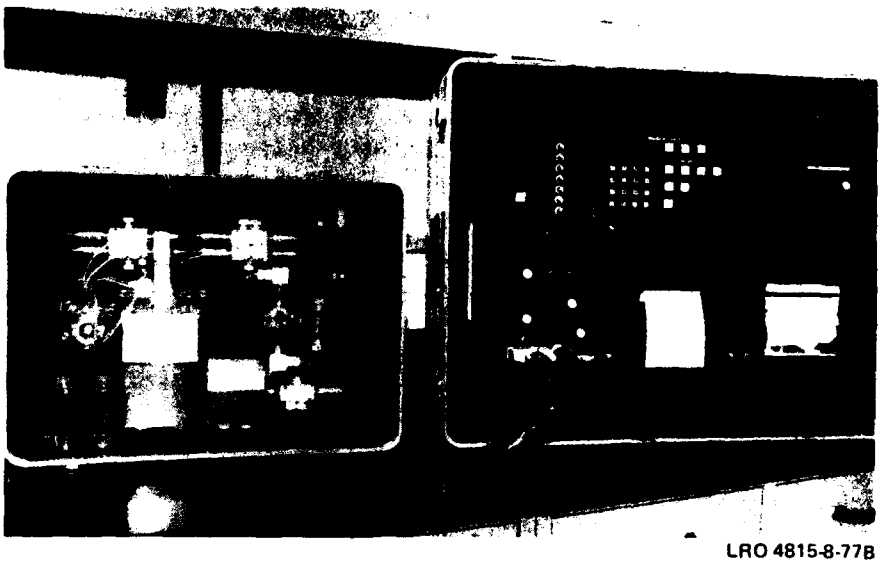
The initial success of the flow-through instrument prompted the design and construction of a new version utilizing a microcomputer. The decision to use a microcomputer was based on experience with the previous instrument, which clearly showed the great need for flexibility in such a system in order to cope with various environments in which the instrument was used as well as to take advantage of new analytic methodologies that might become available. Also, as the system evolved and more equipment was added, such as pH meters, specific ion meters, and fluorometers, the amount of raw data generated became unwieldy, requiring months of processing following a sea cruise.

The use of a microcomputer as both system controller and data manipulator facilitated data handling and increased flexibility. The microcomputer permits a great deal of latitude in system hardware configuration. The instrument is now able to incorporate ancillary instruments and coordinate the different functions of the total system. Because programs can be easily changed, modified, or updated, the instrument hardware package can evolve fresh capabilities as new equipment, sensors, or techniques are developed. An example of such a modification would be the addition of an auxiliary reservoir and associated valves to make it possible to alter the medium present in the electrodes during the scanning cycle. This procedure for stripping or scanning into a different electrolyte permits analysis of metals other than Cu, Pb, Cd, and Zn. For example, by stripping into a $HClO_4$ solution, the instrument can be used for the detection of mercury in seawater (Ref. 7).

A further benefit derived from use of a microcomputer is the powerful data processing capability of the instrument. Thus, it is now possible to achieve real-time data reduction, permitting parameter adjustments to the system based on the processed data displayed.

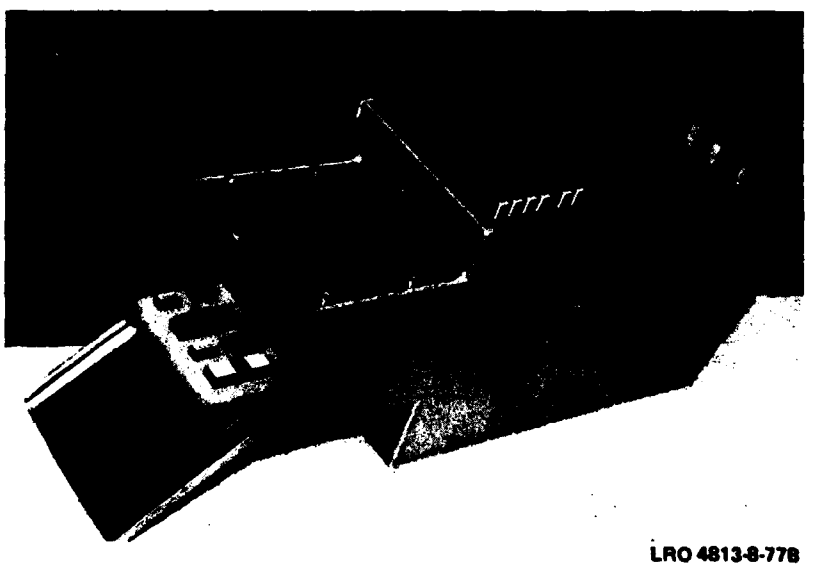
HARDWARE

To facilitate alterations to the system, the microcomputer section in the instrument has been made readily accessible to the user. Thus, there are two major functional modes of operation: the dedicated ASV instrument system, with its programs on EPROMS (erasable programmable read-only memories), and an accessible, independent microcomputer. The current electronic configuration (Figs. 7 and 8) utilizes a Motorola 6800 microcomputer with 16K words of RAM (random-access memory) and sockets for 32K words of EPROMS. The control functions are implemented through eight peripheral interface adaptors with a total of 128 I/O lines, which can be programmed as inputs or outputs, and 32 interrupt lines, half of which can be programmed as inputs or outputs. Interfacing to high-current components such as motors and solenoids is effected through solid state relays and transistor drivers. Two independent clocks are available. One is used for time of day as well as some timing sequences, while the other is a digital-panel-meter elapsed timer that can be programmed to display any desired number of seconds and countdown, giving an output pulse as the zero time indication. The elapsed time provides the operator with a visual countdown, allowing him to monitor the time remaining during any cycle of a sequence as well as signalling the computer when the timing is complete. Data input to the computer is via a 16-key keyboard. Experimental parameters, such as cycle times and voltage values, are input in decimal form and converted to BCD (binary coded decimal) and binary formats for machine use. Computed results are presented on a 40-column alphanumeric printer, while the raw data is printed on a 5- or 10-in. strip chart recorder. See Appendix B for complete system schematics.



LRO 4815-8-77B

Figure 7. Microcomputer-assisted flow-through ASV system.



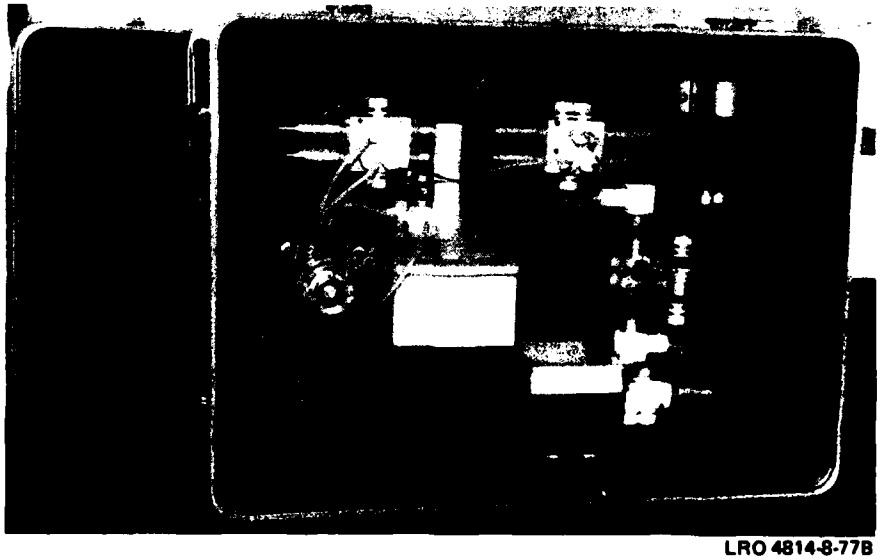
LRO 4813-8-77B

Figure 8. Microcomputer component of new flow-through ASV system.

The chemical cell, as shown in Fig. 9, consists of five custom Teflon solenoid valves, sample and mercury reservoirs, pumps, and the necessary plumbing. Fig. 10 is a diagrammatic representation of the chemical package. It illustrates the interconnections between the valves, pumps and reservoirs. Fig. 10 should be used for reference in the following example of a single analysis sequence. The sequence begins by switching valves 1 and 2 to the A site and activating pump number 1, causing the mercury solution to be circulated through the electrodes.

A negative potential (vs the reference) is now applied to the carbon working electrode for approximately 6 s, causing a thin film of mercury to be plated on the carbon electrode's surface. At the end of this time, valves 1 and 2 are switched to the B side and the sample is now pumped through the electrodes. The sample may either be recirculated or discarded, depending on the setting of valve 3. In this example, the sample is discarded after making one pass through the electrodes. The potential applied previously to the mercury solution is maintained, and the 1-liter sample is plated for 4 min. When the sampling is completed, pump 1 deactivates, and the potential scan of the electrode begins. The peak current data collected at this time is stored in memory and is processed during the next sequence. Concurrently, the sample reservoir is flushed and refilled with a fresh sample. At the conclusion of the scanning cycle, acid or standard may be added to the sample if required. Finally pump 1 is reactivated, valves 1 and 2 are switched to the A side, and the entire sequence is repeated following a delay to purge oxygen from the new sample.

Although this is a very brief description of the actual analysis procedure, the important point to consider is the flow-through nature of the electrodes and the relatively simple plumbing needed to automatically control the analysis. This plumbing can be altered or extended to provide more capability, such as the addition of auxiliary reservoirs, as indicated in Fig. 10, should the need arise.



LRO 4814-8-77B

Figure 9. Chemical cell for new flow-through ASV system.

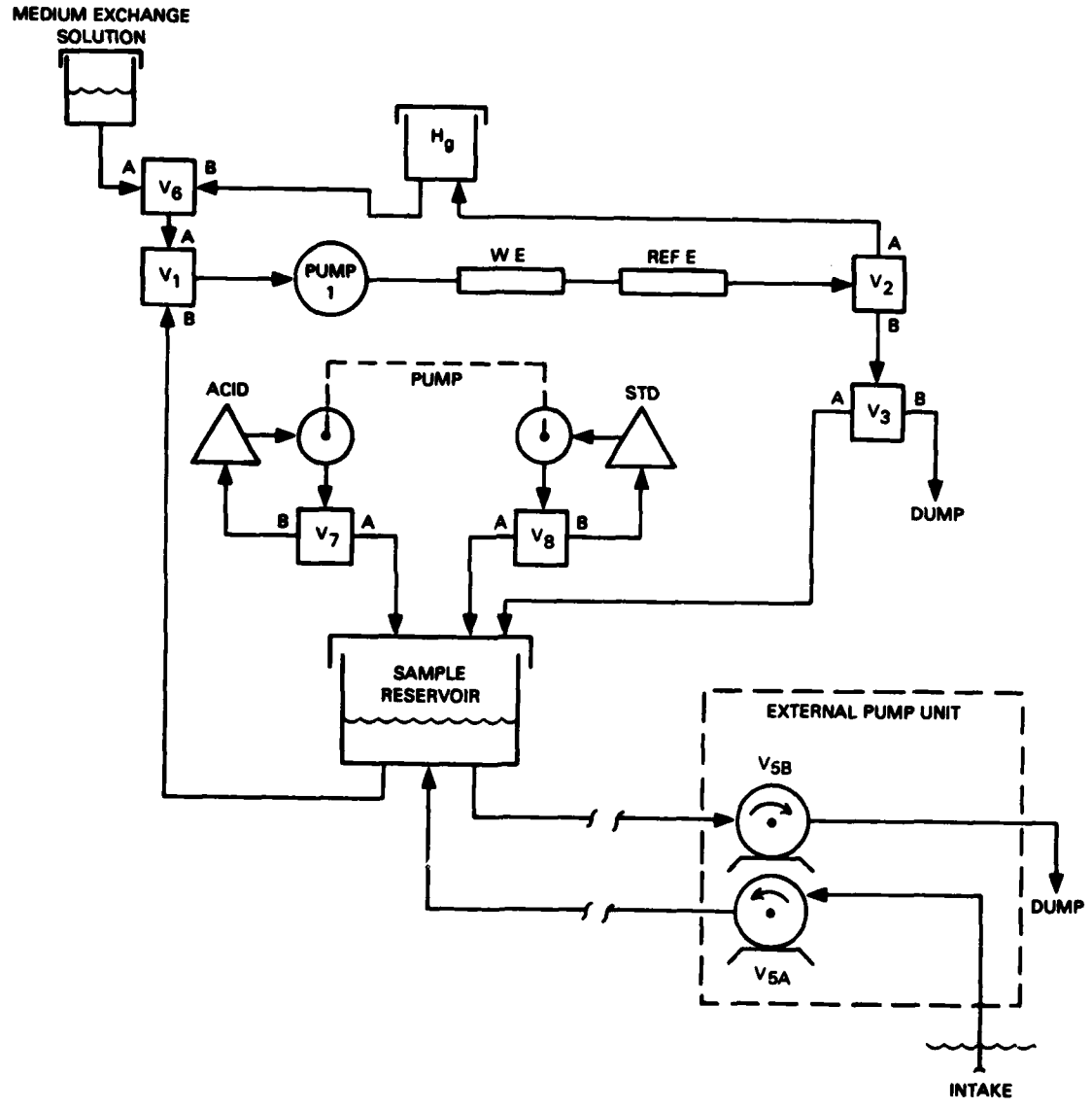


Figure 10. Chemical package plumbing schematic.

SOFTWARE

The software developed for the microcomputer consists of two major system routines. The first is the controlling program for the ASV instrumentation. This program walks the operator through the initialization of the system by the use of the interactive questions and answers. The current program asks 28 questions covering all system parameter settings as well as the different operating mode options. Any of these inputs may be changed by the operator at any time during system operation. Once the instrument has been initialized, the start command is given by pushing the "GO" button and the analyses will proceed without further operator intervention. To allow system monitoring, critical parameters such as motor status, valve status, cycle status, and electrode voltage, which could affect an analysis, are continually displayed on the front panel via LEDs or digital panel meters.

The second major system program is the computing routine. This program operates on the stored digitized data, calculates the actual metal concentrations in ppb, and prints the results along with sample number, date, and sample acquisition time on the 40-column printer. The program operates on the digitized data generated during the potential scan. It automatically locates the current peaks for each metal and determines the validity of each peak by comparison with the programmed voltage range. The ranges are entered by the operator during initialization but can be changed at any time. Once the peaks have been located, their areas are calculated by the trapezoid rule for stepwise integration. The area values are now stored until the areas of the sample plus standard have also been calculated.

When the areas for both the samples and the sample plus standard have been determined, the program calculates the actual metal ion concentration by means of the following equation:

$$C_u = C_s \cdot \left(\frac{A_u}{A_{su} - A_u} \right)$$

Where

C_u = Concentration of unknown metal ion in ppb

C_s = Concentration of standard in ppb

A_u = Peak area for metal ion

A_{su} = Peak area for metal ion + standard addition

The results are then printed and the memory purged for the next set of data. A sample of the printout is shown in Fig. 11. A complete listing of all software is found in Appendix A.

As noted, the system's microcomputer module is accessible to the investigator and can be used for further program development. Provisions exist to allow the attachment of a CRT terminal and a dual floppy-disk mass-storage unit. When used in this fashion and with substitution of a simulator in place of the wet-chemistry electromechanical unit, the system becomes a powerful development tool, enabling new software and hardware designs to be developed and evaluated (including I/O simulation) without the need to invest in new equipment before a design is proven. The simulator also doubles as a system test and calibration

```
ENTER  YEAR
       YEAR = 1979.
ENTER  JULIAN DATE
       JULIAN DATE = 196.

SET REAL-TIME CLOCK TO CURRENT TIME:
    THEN PUSH "GO"


***  SELECT PROGRAM MODE  ***


ENTER INITIAL POTENTIAL
      INITIAL POTENTIAL = -1.2
ENTER FINAL POTENTIAL
      FINAL POTENTIAL = -.12


     * * * * * * * * * * * * * * *
     * VARIABLE PARAMETER MODE *
     * * * * * * * * * * * * * * *


  * ENTER THE FOLLOWING DATA *

ENTER HG PLATTING TIME (MIN&SEC)
      HG PLATTING TIME (MIN&SEC) = 7.

ENTER SAMPLE PLATE TIME(MIN&SEC)
      SAMPLE PLATE TIME(MIN&SEC) = 430.

ENTER VALVE DELAY1 (SEC)
      VALVE DELAY1 (SEC) = 2.

ENTER VALVE DELAY2 (SEC)
      VALVE DELAY2 (SEC) = 2.

ENTER SCAN TIME (MIN&SEC)
      SCAN TIME (MIN&SEC) = 145.

ENTER FLUSHING TIME (SEC)
      FLUSHING TIME (SEC) = 50.

ENTER STD. ADDITION TIME (SEC)
      STD. ADDITION TIME (SEC) = 8.

ENTER ACID ? (YES=1,NO=0)
      ACID ? (YES=1,NO=0) = 1.

ENTER ACID ADDITION TIME (SEC)
      ACID ADDITION TIME (SEC) = 8.

ENTER PURGE DELAY TIME (MIN&SEC)
      PURGE DELAY TIME (MIN&SEC) = 200.

ENTER ZINC ANALYSIS (YES=1,NO=0)
      ZINC ANALYSIS (YES=1,NO=0) = 0.

ENTER CYCLE # FOR STD. ADD .
      CYCLE # FOR STD. ADD . = 3.

ENTER  CD STANDARD CONC.-IN PPB
       CD STANDARD CONC.-IN PPB =  .2

ENTER  PB STANDARD CONC.-IN PPB
       PB STANDARD CONC.-IN PPB =  .4

ENTER  CU STANDARD CONC.-IN PPB
       CU STANDARD CONC.-IN PPB = 1.

ENTER CD LOWER LIMIT (IN VOLTS)
      CD LOWER LIMIT (IN VOLTS) =  .9

ENTER CD UPPER LIMIT (IN VOLTS)
      CD UPPER LIMIT (IN VOLTS) =  .5

ENTER PB LOWER LIMIT (IN VOLTS)
      PB LOWER LIMIT (IN VOLTS) =  .7

ENTER PB UPPER LIMIT (IN VOLTS)
      PB UPPER LIMIT (IN VOLTS) =  .3

ENTER CU LOWER LIMIT (IN VOLTS)
      CU LOWER LIMIT (IN VOLTS) =  .57

ENTER CU UPPER LIMIT (IN VOLTS)
      CU UPPER LIMIT (IN VOLTS) =  .13


 IF THE POTENTIOSTATE PARAMETERS
   ARE SET, PUSH  "GO"


DATA OUTPUT  ** CONCENTRATIONS IN PPB **


*001    1979  196  1444:33
     CU:    .00    PB:    .00    CD:    .00


*002    1979  196  1451:01
     CU:    .00    PB:    .00    CD:    .00
```

Figure 11. Sample printout.

unit for use in the field or laboratory, permitting a quick and safe determination of proper instrument function prior to actual use.

The ASV system's capacity for expansion or modification is limited only by the physical constraints, such as size and weight, that the designer might impose and by the total available memory, both core and mass storage. With the advent of bubble and CCD solid state mass-storage memories, it would be possible to combine very large amounts of memory in relatively small packages, facilitating further expansion of system capability with only a modest increase in size and weight.

FUTURE DEVELOPMENTS

Work is currently underway to characterize a new electrode material that has shown great promise. The material is a low-temperature isotropic carbon in the form of a thin (1/16-in.) disk with a highly polished surface. These inexpensive disks can be used as received and are easily reconditioned by light polishing with fine diamond abrasive. Preliminary results indicate this electrode should function continuously for at least 24 hours before requiring reconditioning.

Investigations into what parameters affect sensitivity and a determination of the electrodes' actual useful lifetime will be carried out during the Verifront Equatorial Cruise of November 1979. This cruise will also provide the first extensive field test of the entire computer-controlled ASV system. Following the Verifront Project, the system will undergo continuous testing and evaluation in San Diego Bay throughout 1980, concurrent with the writing of a comprehensive operations manual.

These lengthy field tests will provide the necessary operation time to enable us to optimize the system's software and hardware configurations.

REFERENCES

1. E. Barendrecht, Electroanalytical Chemistry, Marcel Dekker, New York, N.Y., Vol. 2, 1967.

2. W.D. Ellis, "Anodic Stripping Voltammetry," J. Chem. Ed., 50 A131-A147 (1973).

3. A. Zirino, S. Lieberman, and M.L. Healy. "Anodic Stripping Voltammetry of Trace Metals in Seawater," in Marine Electrochemistry (J.B. Berkowitz, R.A. Horne, M. Banus, P.L. Howard, M.J. Pryor and A.G. Whitnak, eds.), Electrochem. Soc., Princeton, N.J. 1973, pp. 319-332.

4. M. Whitfield, Chemical Oceanography, Academic Press, New York, N.Y., 1975.

5. S. Liberman, and A. Zirino. Anodic "Stripping Voltammetry of Zinc in Seawater with a Tubular Mercury-Graphite Electrode," Anal. Chem. 6, 20-23, (1974).

6. A. Zirino, S. Lieberman and C. Clavell, "Measurement of Copper and Zinc in San Diego Bay by Automated Anodic Stripping Voltammetry," Envir. Sci. Technol., 12, 73 (1978).

7. R. Fukai, and L. Huynh-Ngoc, "Direct Determination of Mercury in Seawater by Anodic Stripping Voltammetry with a Graphite Electrode," Anal. Chem. Acta., 83 375-379 (1975).

INDEX TO APPENDIX
A

SYSTEM CONTROL AND COMPUTING PROGRAMS

All software was written in 6800 assembly language under Motorola's EDOS Operating system, using Motorola's Macro Assembler and Linking Loader.

Control Prog Size = ~8K Computing Prog = ~4K

SECTION I: Utility Subroutines

Label	Function	Page
BCDB	BCD-to-Binary Conversion	19
RMUL	16 X 16 BIT multiply	21
DV16	16 X 16 BIT Divide	22
DIV3	32 X 16 BIT Divide	24
TIME	Real-time clock display	26
PRT1	Message Printer	27
CLK	Real-time clock input handler	28
GNTM	General-Purpose Timing Loop	32

SECTION II: Keyboard interface and A/D–D/A set-up subroutines

Label	Function	Page
FRTQ	A/D and D/A set-up	35
KBIN	Keyboard input and conversion	43
PRNT	Printer controller	49
DSPL	LED display	52

SECTION III: Main System Control Routines

Label	Function	Page
RESET	System initialization	55
INIT	Parameter set-up	62
PROG	Main system control program	73
MESG	Message list	86
POLL	Interrupt polling	88
SCAN	Data collecting	94
HALT	Parameter changing	101
QFIX	Question repeat	103
CT	Toggle for elapsed timer	104

SECTION IV: Computing Routine

Label	Function	Page
COMP	Data reduction and concentration calculations	107

SECTION I:
Utility Subroutines

```
001   BCDB   ***  BCD TO BINARY CONVERSION ROUTINE  ***

                        NAM     BCDB      VER.1.1      11-22-77      C
                    *
                    *  FILE NAMES:   &BCDB (S) / BCDB% (R) / BCDB (A)
                    *
                        OPT     REL
                    *
                        TTL     ***  BCD TO BINARY CONVERSION ROUTI
                    *
                    ************************************************
                    *
                    *    THIS SUBROUTINE CONVERTS 4 PACKED BCD DIGITS
                    *    16 BIT BINARY NUMBER.
                    *
                    *    INPUT IS IN A (MSB) & B (LSB) REG
                    *   OUTPUT IS IN A (MSB) & B (LSB)
                    ************************************************
                    *
                    *
0000 B7 0000   D BCDBIN STAA    SAVE1
0003 7F 0001   D        CLR     BINUPR
0006 17                 TBA
0007 C4 0F     A        ANDB    #$0F
0009 44                 LSRA
000A 44                 LSRA
000B 44                 LSRA
000C 44                 LSRA
000D 27 05 0014 TENLP   BEQ     DOHUND
000F CB 0A     A        ADDB    #10
0011 4A                 DECA
0012 20 F9 000D         BRA     TENLP
0014 0C         DOHUND  CLC
0015 B6 0000   D        LDAA    SAVE1
0018 84 0F     A        ANDA    #$0F
001A 27 0A 0026 HUNLP   BEQ     DOTHOU
001C CB 64     A        ADDB    #100
001E 24 03 0023         BCC     HUNOO
0020 7C 0001   D        INC     BINUPR
0023 4A         HUNOO   DECA
0024 20 F4 001A         BRA     HUNLP
0026 B6 0000   D DOTHOU LDAA    SAVE1
0029 44                 LSRA
002A 44                 LSRA
002B 44                 LSRA
002C 44                 LSRA
002D B7 0000   D        STAA    SAVE1
0030 26 05 0037         BNE     THOU00
0032 B6 0001   D        LDAA    BINUPR
0035 20 0D 0044         BRA     XITBIN
0037 B6 0001   D THOU00 LDAA    BINUPR
003A 0C         THOULP  CLC
003B CB E8     A        ADDB    #232
003D 89 03     A        ADCA    #$03
003F 7A 0000   D        DEC     SAVE1
0042 26 F6 003A         BNE     THOULP
0044 39         XITBIN  RTS
                    *
                    *
                        XDEF    BCDBIN
```

```
002   BCDB    ***  BCD TO BINARY CONVERSION ROUTINE  ***

                   *
0000                       DSCT
                   *
0000     0001  A SAVE1  RMB     1
0001     0001  A BINUPR RMB     1
                   *
                           END

ERRORS 00000
```

```
001   RMULT  ***  16 X 16 BIT MULTIPLY  ***

                         NAM    RMULT
                     *
                     *  FILE NAMES:   &RMUL (S) / RMUL (R)
                     *
                         OPT    REL
                     *
                         TTL    ***  16 X 16 BIT MULTIPLY  ***
                     *
                     ************************************
                     *    UNSIGNED MULTIPLY ROUTINE
                     *
                     *   TO SET UP FOR USE:
                     *  LDAA  (HI BYTE)    16 BIT VALUE
                     *  LDAB  (LO BYTE)    MULTIPLICAN
                     *  INS
                     *  INS
                     *  INS
                     *  INS
                     *  INS
                     *  PSHB
                     *  PSHA
                     *  LDAA    (HI BYTE)  MULTIPLIER
                     *  LDAB  (LO BYTE)
                     *  PSHB
                     *  PSHA
                     *  DES
                     *  JSR MULT16
                     *
                     *   RETURNS WITH HI BYTE IN A REG
                     *     & LO BYTE IN B REG
                     *
                     ************************************
                     *
                     *
0000 30              MULT16 TSX
0001 86 10     A            LDAA   #16
0003 A7 02     A            STAA   2,X
0005 4F                     CLRA
0006 5F                     CLRB
0007 66 03     A            ROR    3,X
0009 66 04     A            ROR    4,X
000B 24 04 0011 NNEXT  BCC    RROTN
000D EB 06     A            ADDB   6,X
000F A9 05     A            ADCA   5,X
0011 46              RROTN  RORA
0012 56                     RORB
0013 66 03     A            ROR    3,X
0015 66 04     A            ROR    4,X
0017 6A 02     A            DEC    2,X
0019 26 F0 000B             BNE    NNEXT
001B 39                     RTS
                     *
                     *    EXTERNAL DEFINITIONS
                     *
                         XDEF   MULT16
                     *
                         END

ERRORS 00000
```

```
001   DV16   ***  16 X 16 BIT DIVIDE ROUTINE  ***

                       NAM    DV16     VER.1.1    11-22-77   CLAV
                  *
                  *  FILE NAMES:   SDV16 (S) / DV16 (R)
                  *
                       OPT    REL
                  *
                       TTL    ***  16 X 16 BIT DIVIDE ROUTINE  **
                  *
                  ************************************************
                  *
                  *    THIS ROUTINE DIVDES TWO UNSIGNED 16 BIT
                  *    NUMBERS INTO EACH OTHER AND RETURNS A
                  *     16 BIT ANSWER.
                  *
                  *  TO USE:  ENTER WITH HI BYTE OF DIVIDEND IN A
                  *             & LO BYTE IN B
                  *          ADDRS. OF HI BYTE OF DIVISOR IN X
                  *
                  *       RETURNS:  A - HI BYTE RESULT
                  *                 B - LO BYTE RESULT
                  *
                  ************************************************
                  *
                  *
                  *
0000 37           DIV16  PSHB
0001 36                  PSHA
0002 A6 00     A         LDAA   X
0004 E6 01     A         LDAB   1,X
0006 37                  PSHB
0007 36                  PSHA
0008 34                  DES
0009 30                  TSX
000A 86 01     A         LDAA   #1
000C 6D 01     A         TST    1,X
000E 2B 0B 001B          BMI    DIV153
0010 4C           DIV151 INCA
0011 68 02     A         ASL    2,X
0013 69 01     A         ROL    1,X
0015 2B 04 001B          BMI    DIV153
0017 81 11     A         CMPA   #17
0019 26 F5 0010          BNE    DIV151
001B A7 00     A  DIV153 STAA   X
001D A6 03     A         LDAA   3,X
001F E6 04     A         LDAB   4,X
0021 6F 03     A         CLR    3,X
0023 6F 04     A         CLR    4,X
0025 E0 02     A  DIV163 SUBB   2,X
0027 A2 01     A         SBCA   1,X
0029 24 07 0032          BCC    DIV165
002B EB 02     A         ADDB   2,X
002D A9 01     A         ADCA   1,X
002F 0C                  CLC
0030 20 01 0033          BRA    DIV167
0032 0D           DIV165 SEC
0033 69 04     A  DIV167 ROL    4,X
0035 69 03     A         ROL    3,X
0037 64 01     A         LSR    1,X
```

```
002   DV16    ***  16 X 16 BIT DIVIDE ROUTINE  ***

 0039 66 02     A          ROR     2,X
 003B 6A 00     A          DEC     X
 003D 26 E6 0025           BNE     DIV163
 003F 31                   INS
 0040 31                   INS
 0041 31                   INS
 0042 32                   PULA
 0043 33                   PULB
 0044 39                   RTS
                  *
                  *
                            XDEF    DIV16
                  *
                            END
ERRORS 00000
```

```
001   DIV3   ***  32 X 16 BIT DIVIDE  ***

                         NAM    DIV3     VER. 1.1   11-22-77   CLAV
                     *
                     *  FILE NAMES:    &DIV3 (S) / DIV3 (R)
                     *
                         OPT    REL
                     *
                         TTL    ***  32 X 16 BIT DIVIDE  ***
                     *
                     ********************************************************
                     *
                     *    THIS ROUTINE DIVIDES A 16 BIT NUMBER INTO A 32 BIT
                     *    NUMBER AND RETURNS A 16 BIT NUMBER.
                     *
                     ********************************************************
                     *
                     *
                     *
0000 8D 4A 004C DIV32  BSR    OVFTST
0002 CE 0011  A L5     LDX    #17
0005 5F         L0     CLRB
0006 B6 0004  D        LDAA   DSORU
0009 B1 0001  D        CMPA   DENDHU
000C 27 30 003E        BEQ    L2
000E 22 13 0023        BHI    L1
0010 5C         L3     INCB
0011 B6 0000  D        LDAA   DENDHL
0014 B0 0005  D        SUBA   DSORL
0017 B7 0000  D        STAA   DENDHL
001A B6 0001  D        LDAA   DENDHU
001D B2 0004  D        SBCA   DSORU
0020 B7 0001  D        STAA   DENDHU
0023 8C 0001  A L1     CPX    #1
0026 27 20 0048        BEQ    L7
0028 8D 39 0063        BSR    STSHF
002A 79 0000  D        ROL    DENDHL
002D 79 0001  D        ROL    DENDHU
0030 FA 0003  D L6     ORAB   DENDLL
0033 F7 0003  D        STAB   DENDLL
0036 09                DEX
0037 26 CC 0005        BNE    L0
0039 FE 0002  D        LDX    DENDLU
003C 0A                CLV
003D 39                RTS
003E B6 0005  D L2     LDAA   DSORL
0041 B1 0000  D        CMPA   DENDHL
0044 22 DD 0023        BHI    L1
0046 20 C8 0010        BRA    L3
0048 8D 19 0063 L7     BSR    STSHF
004A 20 E4 0030        BRA    L6
004C B6 0001  D OVFTST LDAA   DENDHU
004F B1 0004  D        CMPA   DSORU
0052 2E 0B 005F        BGT    OVFYES
0054 2D 08 005E        BLT    OVFNO
0056 B6 0005  D        LDAA   DSORL
0059 B1 0000  D        CMPA   DENDHL
005C 23 01 005F        BLS    OVFYES
005E 39         OVFNO  RTS
005F 32         OVFYES PULA
```

```
002   DIV3   ***  32 X 16 BIT DIVIDE  ***

 0060 32                    PULA
 0061 0B                    SEV
 0062 39                    RTS
 0063 78 0003  D STSHF      ASL     DENDLL
 0066 79 0002  D            ROL     DENDLU
 0069 39                    RTS
                   *
                   *
                            XDEF    DENDHU,DENDHL,DENDLU,DENDLL
                            XDEF    DSORU,DSORL,DIV32
                   *
 0000                       DSCT
                   *
 0000    0001  A DENDHL     RMB     1
 0001    0001  A DENDHU     RMB     1
 0002    0001  A DENDLU     RMB     1
 0003    0001  A DENDLL     RMB     1
 0004    0001  A DSORU      RMB     1
 0005    0001  A DSORL      RMB     1
                   *
                            END
ERRORS 00000
```

```
001   TIME   ***  REAL-TIME LED DISPLAY SUBROUTINE  ***

                        NAM    TIME     VER. 2   4-9-78     CLAVEL
                    *
                    *         FILE NAME:   &TIME (S)/ TIME (R)
                    *
                        OPT    REL
                    *
                        TTL    ***  REAL-TIME LED DISPLAY SUBROUTI
                    *
                    ************************************************
                    *
                    *   THIS ROUTINE DISPLAYS THE CURRENT TIME ON THE
                    *   LED'S UNTIL THE "GO" BUTTON IS PUSHED.  TIME
                    *   APPROX. EVERY SEC..
                    *
                    *   THE ROUTINE IA A SUBROUTINE AND WILL ONLY BE
                    *   DURING INTIALIZATION TO ALLOW SETTING OF THE
                    *
                    *   "GO" COMES IN ON : CB1-PIA5
                    *
                    ************************************************
                    *
0000 B6 0000  A TIME   LDAA   P5BP     DUMMY READ TO CLR
                    *                  INTER. FLAGS
0003 7C 0000  A        INC    TIMFLG   SET TIMFLG
0006 BD 0000  A SKIP76 JSR    CLOCK    GET TIME AND DIPLAY
                    *
0009 B6 0000  A        LDAA   P5BC     CHECK IF "GO" PUSHED
000C 2B 02 0010        BMI    SKIP75   YES- RTS
                    *
000E 20 F6 0006        BRA    SKIP76   NO- READ CLOCK AGAIN
                    *
0010 7F 0000  A SKIP75 CLR    TIMFLG   CLR FLAG
0013 39                RTS             RETURN
                    *
                    *
                    *    ***  EXTERNAL REFERENCES  ***
                    *
                        XREF   TIMFLG,CLOCK,P5BC,P5BP
                    *
                    *
                    *    ***  EXTERNAL DEFINITIONS  ***
                    *
                        XDEF   TIME
                    *
                        END
ERRCRS 00000
```

```
001   PRINT1 ***  MESSAGE PRINTING SUBROUTINE ***

                                NAM     PRINT1   VER. 1    11-8-77    CLAVE
                          *
                          *   FILE NAMES:  &PRT1 (S) /PRT1 (R)
                          *
                                OPT     REL
                          *
                                TTL     *** MESSAGE PRINTING SUBROUTINE **
                          *
                          *
0005                              ORG     $0005
                          *
0005     0002   A STRADR  RMB     2
                          *
0000                              PSCT
                          *
0000 DF 05     A PRINT1   STX     STRADR   MESS BUFF STARTING ADDR.
0002 B6 0000   D          LDAA    BLOCK    # OF LINES IN MESS.
0005 B7 0007   A          STAA    BLKADR
                          *
0008 BD B2C3   A          JSR     SUB1     PRINT BUFF
000B BD B2A0   A          JSR     SUB2
                          *
000E 39                   RTS
                          *
                          *
                          *
                                XDEF    SUB1,SUB2,SUB3,BLOCK,BLKADR,STRADR,
                          *
         B2C3  A SUB1     EQU     $B2C3
         B2A0  A SUB2     EQU     $B2A0    DRIVER
         B38B  A SUB3     EQU     $B38B    LF
                          *
         0007  A BLKADR   EQU     $0007
                          *
0000                              DSCT
                          *
0000     0001  A BLOCK    RMB     1
                          *
                                  END
ERRORS 00000
```

```
001   CLOCK  ***  REAL-TIME CLOCK INPUT SUBROUTINE  ***

                          NAM    CLOCK    VER. 3   4-18-79    CLAVEL
                    *
                    *     FILE:  &CLK (S)/ CLK (R)
                    *
                          OPT    REL
                    *
                          TTL    ***  REAL-TIME CLOCK INPUT SUBROUTI
                    *
                    *
                    ************************************************
                    *    THIS SUBROUTINE READS IN THE CURRENT TIME
                    *    FROM THE REAL-TIME CLOCK IN BCD DIGITS (S),
                    *    ONE AT A TIME, BY TOGGLING THE CB2(PIA7)
                    *    LINE TO ADVANCE THE DIGIT INPUT.  THE
                    *    DIGITS ARE STORED IN (TBCDB1-8) AND ALSO
                    *    THE HR,MIN & SEC DIGITS ARE PACKED IN
                    *    TBCDE,TBCDM & TBCDS.
                    *
                    *    THIS ROUTINE IS ALSO USED BY THE RESET
                    *    PROG TO DISPLAY THE TIME ON THE LED'S TO
                    *    ALLOW SETTING THE CLOCK.
                    *
                    ************************************************
                    *
                    *
0000                      DSCT
                    *
0000    0002  A DIGITU RMB    2
0002    0001  A TBCDH  RMB    1
0003    0001  A TBCDM  RMB    1
0004    0001  A TBCDS  RMB    1
0005    0001  A THRB   RMB    1
0006    0002  A TBCDBU RMB    2
0008    0001  A TSTORU RMB    1
0009    0001  A TSTORL RMB    1
000A    0001  A DISPCN RMB    1
                    *
                    *
0200                      PSCT
                    *
                    *
0000 0F           CLOCK  SEI              SET INTERRUPT MASK
0001 B6 0000  A          LDAA   P7BP      READ B SIDE OF PIA
0004 43                  COMA             COMPLEMENT A
                    *                   BIT 7 SET?  STATR WITH SEC DI
0005 2B 1C 0023          BMI    SKIP70    YES- GO TO SKIP70
                    *                   OTHERWISE TOGGLE CB2-PIA7
0007 B6 0000  A          LDAA   P7BC
000A 8B 08    A          ADDA   #$08      SET BIT 3
000C B7 0000  A          STAA   P7BC
                    *
000F C6 0F    A          LDAB   #15       100 US DELAY
0011 5A         STALL    DECB
0012 26 FD 0011          BNE    STALL
                    *
0014 B6 0000  A          LDAA   P7BC
0017 84 F7    A          ANDA   #$F7      CLR BIT 3
0019 B7 0000  A          STAA   P7BC
```

```
002   CLOCK  ***  REAL-TIME CLOCK INPUT SUBROUTINE  ***

                      *
001C 86 C8     A      LDAA   #200
001E 4A        STAL2  DECA                1 MS DELAY
001F 26 FD 001E       BNE    STAL2
                      *
0021 20 DD 0000       BRA    CLOCK       START OVER
                      *
0023 C6 06     A SKIP70 LDAB   #6         SET UP COUNTER
0025 CE 000B   D      LDX    #TBCDB1     LOAD X WITH 1ST BUFF ADDR.
0028 84 0F     A SKIP71 ANDA   #$0F        MASK OFF UPPER 4 BITS
                      *
002A A7 00     A      STAA   X           PUT DIGIT INTO BUFF
                      *
002C B6 0000   A      LDAA   P7BC         ON
002F 8B 08     A      ADDA   #$08         TOGGLE CB2 FOR
0031 B7 0000   A      STAA   P7BC         NEXT DIGIT
                      *
0034 86 0F     A      LDAA   #15
0036 4A        STAL1  DECA                100μS
0037 26 FD 0036       BNE    STAL1       DELAY
                      *
0039 B6 0000   A      LDAA   P7BC
003C 84 F7     A      ANDA   #$F7         OFF CB2
003E B7 0000   A      STAA   P7BC
                      *
0041 5A               DECB               DONE 6 TIMES?
0042 27 0C 0050       BEQ    SKIP72      YES- GO TO SKIP72
                      *                  NO- CONTINUE
0044 08               INX                INC THE BUFF ADDR.
                      *
0045 86 FA     A      LDAA   #250
0047 4A        STAL3  DECA                1.5 MS DELAY
0048 26 FD 0047       BNE    STAL3
                      *
004A B6 0000   A      LDAA   P7BP         READ CLOCK
004D 43               COMA               COMPLEMENT A
004E 20 D8 0028       BRA    SKIP71      LOOP
                      *
0050 7D 0000   A SKIP72 TST    TIMFLG     CALLED FROM TIME?
0053 26 29 007E       BNE    SKIP73      YES- GO TO SKIP73
                      *
                      *                     OTHERWISE CONTINUE
0055 B6 000C   D      LDAA   TBCDB2
0058 48               ASLA
0059 48               ASLA
005A 48               ASLA
005B 48               ASLA                      (SEC'S DIGITS
005C BB 000B   D      ADDA   TBCDB1       PACK ~~LOWER 4~~
                      *                  ~~DIGITS~~ INTO TBCDS
005F B7 0004   D      STAA   TBCDS
                      *
0062 B6 000E   D      LDAA   TBCDB4
0065 48               ASLA                      min's
0066 48               ASLA               PACK ~~UPPER 4~~
0067 48               ASLA               DIGITS INTO
0068 48               ASLA               TBCDM
0069 BB 000D   D      ADDA   TBCDB3
                      *
```

```
003   CLOCK  ***  REAL-TIME CLOCK INPUT SUBROUTINE  ***

 006C B7 0003  D         STAA    TBCDM     MIN.'S
                *
 006F B6 0010  D         LDAA    TBCDB6
 0072 48                 ASLA
 0073 48                 ASLA              PACK HR'S
 0074 48                 ASLA              INTO TBCDH
 0075 48                 ASLA
 0076 BB 000F  D         ADDA    TBCDB5
                *
 0079 B7 0002  D         STAA    TBCDH
                *
 007C 0E                 CLI
 007D 39                 RTS               RETURN
                *
                *
 007E 7F 0008  D SKIP73  CLR     TSTORU    CONVERT BCD DIGITS
 0081 C6 06    A         LDAB    #6        TO SEG. CODE FOR
 0083 CE 000B  D         LDX     #TBCDB1   DISPLAY
 0086 FF 0006  D         STX     TBCDBU
 0089 CE 0000  A         LDX     #DIGIT6
 008C FF 0000  D         STX     DIGITU    TBCDB1 = 1'S SEC
                *                                TBCDB6 = 10'S HR
                *
 008F FE 0006  D SEGCON  LDX     TBCDBU
 0092 A6 00    A         LDAA    X         TBCDB1 INTO A"
 0094 B7 0009  D         STAA    TSTORL    "A" INTO TSTORL
                *
 0097 FE 0008  D         LDX     TSTORU    TBL + (OFSET) INTO "A
 009A A6 00    A         LDAA    TBL,X
                *
 009C FE 0000  D         LDX     DIGITU    CODED DIGIT INTO DIGIT
 009F A7 00    A         STAA    X
                *
 00A1 FE 0006  D         LDX     TBCDBU
 00A4 08                 INX               INC BCD DIGIT BUFF
 00A5 FF 0006  D         STX     TBCDBU
                *
 00A8 FE 0000  D         LDX     DIGITU    DEX SEG CODED
 00AB 09                 DEX               DIGIT BUFF
 00AC FF 0000  D         STX     DIGITU
                *
 00AF 5A                 DECB              IF NOT DONE 6
 00B0 26 DD 008F         BNE     SEGCON    TIMES LOOP BACK
                *
                *
 00B2 C6 3C    A         LDAB    #60       DISPLAY TIME FOR
 00B4 F7 000A  D         STAB    DISPCN    900 MS ON LED'S
                *
 00B7 BD 0000  A SKIP74  JSR     DISPLA
                *
 00BA 7A 000A  D         DEC     DISPCN
 00BD 26 F8 00B7         BNE     SKIP74
                *
 00BF 0E                 CLI
 00C0 39                 RTS
                *
                *
                *   *** EXTERNAL REFERENCES  ***
```

```
004   CLOCK  ***  REAL-TIME CLOCK INPUT SUBROUTINE  ***

                       *
                             XREF   DISPLA,TBL,DIGIT6,TIMFLG,P7BC,P7BP
                       *
                       *
                       *     ***  EXTERNAL DEFINITIONS  ***
                       *
                             XDEF   TBCDB1,TBCDB2,TBCDB3,TBCDB4,TBCDB5,
                             XDEF   TBCDH,TBCDM,TBCDS,CLOCK
                       *
                       *
 000B                        DSCT
                       *
 000B     0001   A TBCDB1 RMB    1
 000C     0001   A TBCDB2 RMB    1
 000D     0001   A TBCDB3 RMB    1
 000E     0001   A TBCDB4 RMB    1
 000F     0001   A TBCDB5 RMB    1
 0010     0001   A TBCDB6 RMB    1
                       *
                       *
                             END
ERRORS 00000
```

```
001   GENTM  ** GENERAL PURPOSE TIMING ROUTINE **

                          NAM    GENTM    VER. 1  12-15-77  CLAVELL
                      *
                      *  FILE NAMES*  &GNTM (S) / GENTM (R)
                      *
                          OPT    REL
                      *
                          TTL    ** GENERAL PURPOSE TIMING ROUTINE *
                      *
                      ******************************************
                      * TIME DELAY ROUTINE: TO USE LOAD A REG
                      * WITH 2 TIMES THE # OF SEC FOR DELAY
                      * THEN CALL GENTIM: IE. LDAA #12
                      * JSR GENTIM / WOULD GIVE A 6 SEC DELAY
                      ******************************************
                      *
                      *
 0000 CE F423  A GENTIM LDX    #$F423
 0003 09        DECRX  DEX
 0004 26 FD 0003       BNE    DECRX
 0006 CE 2A51  A       LDX    #$2A51
 0009 09        DECX   DEX
 000A 26 FD 0009       BNE    DECX
 000C 4A               DECA
 000D 26 F1 0000       BNE    GENTIM
                      *
 000F 39               RTS
                      *
                      *
                          XDEF   GENTIM
                      *
                          END
ERRORS 00000
```

THIS ROUTINE IS NOT PERFECTLY LINEAR:

INPUT TIME (SEC)	ACTUAL TIME (SEC)
5	5.2
10	10.1
15	15.0
20	19.9
30	29.8
40	39.6
60	59.2

SECTION II:

Keyboard Interface and A/D-D/A Set-Up Subroutines

```
001   FRTQ   ***  IP & FP INPUT ROUTINE  ***

                         NAM    FRTQ     VER 14  9-19-79    CLAVELL
                     *
                     *  FILE NAMES:   &FRTQ (S) / FRTQ (R)
                     *
                         OPT    REL
                     *
                         TTL    ***  IP & FP INPUT ROUTINE  ***
                     *
                     ************************************************
                     *
                     * THIS SUBROUTINE ASKS FOR INPUTS FOR THE SCAN
                     * INITIAL AND FINAL POTENTIALS AND FORMS THE END
                     * POINT COMPARISON VALUE AND THE D/A INITIAL
                     * POTENTIAL VALUE.
                     *
                     ************************************************
                     *
                     *
                     *
                     *****      RANGE CALC. SUBROUTINE      ******
                     *
                     *  CALCULATE HIGH & LOW PEAK RANGE VALUES
                     *  FOR USE IN COMPT. STORE RESULTS INTO
                     *  LOWECO,LOWECO+1,ETC.
                     *  LOWECO HAS MOST NEG VALUE, (START OF PEAK)
                     *
                     *  EXAMPLE RANGE    ZN: -1.2 TO -.9
                     *  VALUES:          CD: -.87  TO -.5
                     *                   PB: -.65 TO -.25
                     *                   CU: - .55 TO -.15
                     *  VOLTAGE * 1000 = INTEGER VALUE (IV)
                     *  I.E.:  .65 = 650 ($028A) = (IV)  FOR PB LOW VA
                     *
                     ************************************************
                     *  (IP - IV) * 10/3 * 410/1000 =A/D BIT COUNTS
                     ************************************************
                     *
0000 8E 0F50   A RNGCAL LDS    #$0F50   RESET STACK PT
0003 B7 0002   D        STAA   TP
0006 F7 0003   D        STAB   TP+1
0009 B6 0018   D        LDAA   IPHI     CALCULATE (IP-IV) FOR
000C F6 0019   D        LDAB   IPLO     CD
000F F0 0003   D        SUBB   TP+1     LOW BYTE
0012 B2 0002   D        SBCA   TP       HI BYTE
                     *
0015 31                 INS
0016 31                 INS
0017 31                 INS
0018 31                 INS
0019 31                 INS
001A 37                 PSHB
001B 36                 PSHA            (IP-IV) * 10
001C 86 00     A        LDAA   #$00
001E C6 0A     A        LDAB   #$0A     MULTIPLIER ON STACK (* 10)
0020 37                 PSHB
0021 36                 PSHA
0022 34                 DES
                     *
```

```
002   FRTQ   ***  IP & FP INPUT ROUTINE  ***

 0023 BD 0000  A         JSR     MULT16   16*16 MULTIPLY -32 BIT RES
                *
 0026 B7 0000  A         STAA    DENDHU
 0029 F7 0000  A         STAB    DENDHL   SET UP FOR 32 BIT DIVISION
 002C EE 03    A         LDX     3,X
 002E FF 0000  A         STX     DENDLU
 0031 86 00    A         LDAA    #$00     DIVISOR (3)
 0033 C6 03    A         LDAB    #$03
 0035 B7 0000  A         STAA    DSORU
 0038 F7 0000  A         STAB    DSORL    ((IP-IV) * 10/3)
                *
 003B BD 0000  A         JSR     DIV32
                *
 003E B6 0000  A         LDAA    DENDLU
 0041 F6 0000  A         LDAB    DENDLL
                *
 0044 31                 INS              ((IP-IV) *10/3) * 410
 0045 31                 INS
 0046 31                 INS
 0047 31                 INS
 0048 31                 INS
 0049 37                 PSHB
 004A 36                 PSHA
 004B 86 01    A         LDAA    #$01     * 410
 004D C6 9A    A         LDAB    #$9A
 004F 37                 PSHB
 0050 36                 PSHA
 0051 34                 DES
                *
 0052 BD 0000  A         JSR     MULT16
                *
 0055 B7 0000  A         STAA    DENDHU
 0058 F7 0000  A         STAB    DENDHL   ((IP-IV) * 10/3) *410/1000
 005B EE 03    A         LDX     3,X
 005D FF 0000  A         STX     DENDLU
                *
 0060 86 03    A         LDAA    #$03     /1000
 0062 C6 E8    A         LDAB    #$E8
 0064 B7 0000  A         STAA    DSORU
 0067 F7 0000  A         STAB    DSORL
                *
 006A BD 0000  A         JSR     DIV32
                *
 006D B6 0000  A         LDAA    DENDLU   A/D BIT COUNTS
 0070 F6 0000  A         LDAB    DENDLL
                *
 0073 8E 0F8E  A         LDS     #$0F8E   STACK BACK AS WAS
 0076 39                 RTS
                *
                *
                *
                *
 0077 BF 0000  D FRSTQ   STS     STK      SAVE STK PTR
 007A 8E 0F50  A FRSTQ1  LDS     #$0F50   RESET STK PTR
                *
 007D 7C 001E  D         INC     FRSTQF   FLAG TO INDICATE INTEG. OR
                *                        DEC. #:USED BY KBIN
 0080 CE 0000  A         LDX     #MES30
```

```
003   FRTQ   ***  IP & FP INPUT ROUTINE  ***

0083 BD 0000  A         JSR     PRINT
                *
0086 86 01    A         LDAA    #1
0088 BD 0000  A         JSR     KBIN      INPUT IP
                *
008B B6 0000  A         LDAA    BINHI     CHK FOR SIGN BIT
008E 2A 02 0092         BPL     ERROR1    IF NOT SET : ERROR
0090 20 03 0095         BRA     A1
0092 7E 026C  P ERROR1  JMP     ERROR
                *
0095 84 7F    A A1      ANDA    #$7F      MASK OFF SIGN BIT
                *
0097 B7 0018  D         STAA    IPHI
009A F6 0000  A         LDAB    BINLO     STORE IP IN IPHI & LO
009D F7 0019  D         STAB    IPLO      12 BIT VALUE
                *
00A0 CE 0000  A         LDX     #MES31    PRINT FP MESSAGE
00A3 BD 0000  A         JSR     PRINT
                *
00A6 86 01    A         LDAA    #1
00A8 BD 0000  A         JSR     KBIN
                *
00AB B6 0000  A         LDAA    BINHI
00AE 2A 35 00E5         BPL     ADDFP     IF FP POS, GO TO ADDFP
00B0 84 7F    A         ANDA    #$7F      OTHERWISE MASK SIGN BIT
                *
00B2 B7 0000  A         STAA    BINHI     STORE FP BACK TO BINHI
00B5 B6 0018  D         LDAA    IPHI
00B8 F6 0019  D         LDAB    IPLO
                *
00BB B1 0000  A         CMPA    BINHI     COMPARE IP TO FP
00BE 27 07 00C7         BEQ     LOWCK
00C0 2D 02 00C4         BLT     ERROR2
00C2 20 0D 00D1         BRA     CONT      IS IP>FP?
00C4 7E 026C  P ERROR2  JMP     ERROR
00C7 F1 0000  A LOWCK   CMPB    BINLO     IF YES GO TO ERROR
00CA 2D 02 00CE         BLT     ERROR3
00CC 20 03 00D1         BRA     CONT
00CE 7E 026C  P ERROR3  JMP     ERROR
                *
00D1 B6 0018  D CONT    LDAA    IPHI
00D4 F6 0019  D         LDAB    IPLO
00D7 F0 0000  A         SUBB    BINLO     DIFF=IP-FP
00DA B2 0000  A         SBCA    BINHI
00DD B7 0016  D         STAA    DIFHI
00E0 F7 0017  D         STAB    DIFLO
00E3 20 0F 00F4         BRA     DIFCK     GO TO DIFCK
                *
00E5 F6 0000  A ADDFP   LDAB    BINLO
00E8 FB 0019  D         ADDB    IPLO      DIFF=IP+FP
00EB B9 0018  D         ADCA    IPHI
                *
00EE B7 0016  D         STAA    DIFHI
00F1 F7 0017  D         STAB    DIFLO     DIF IN DIFHI & LO
                *
00F4 81 0B    A DIFCK   CMPA    #$0B      IS DIFF > 3000 DEC.
00F6 2E 04 00FC         BGT     ERROR4
00F8 27 05 00FF         BEQ     DIFCK1    YES: ERROR
```

```
004   FRTQ   ***  IP S FP INPUT ROUTINE  ***

00FA 20 0C 0108          BRA     CONT1     NO: CONTINUE
00FC 7E 026C  P ERROR4   JMP     ERROR
00FF C1 B8    A DIFCK1   CMPB    #$B8
0101 2E 02 0105          BGT     ERROR5
0103 20 03 0108          BRA     CONT1
0105 7E 026C  P ERROR5   JMP     ERROR
                *
0108 CE 01FC  A CONT1    LDX     #ZERO     D/A ZERO VALUE
010B FF 0014  D          STX     ZERO1
010E 7E 01A4  P          JMP     SKIPZN
                *
                ************************************************
                *  CALCULATE PEAK RANGE VALUES ON X AXIS
                *  CALLED FROM "INIT"
                ************************************************
                *
0111 B6 0006  D RNG      LDAA    CD+2
0114 F6 0007  D          LDAB    CD+3
0117 BD 0000  P          JSR     RNGCAL    CD LOWER LIMIT
011A B7 0002  A          STAA    LOWECO+2
011D F7 0003  A          STAB    LOWECO+3
                *
0120 B6 0004  D          LDAA    CD
0123 F6 0005  D          LDAB    CD+1
0126 BD 0000  P          JSR     RNGCAL    CD UPPER LIMIT
0129 B7 0002  A          STAA    HIECO+2
012C F7 0003  A          STAB    HIECO+3
                *
012F B6 000A  D          LDAA    PB+2
0132 F6 000B  D          LDAB    PB+3
0135 BD 0000  P          JSR     RNGCAL    PB LOW
0138 B7 0004  A          STAA    LOWECO+4
013B F7 0005  A          STAB    LOWECO+5
                *
013E B6 0008  D          LDAA    PB
0141 F6 0009  D          LDAB    PB+1
0144 BD 0000  P          JSR     RNGCAL    PB HI
0147 B7 0004  A          STAA    HIECO+4
014A F7 0005  A          STAB    HIECO+5
                *
014D 7D 0000  A          TST     ZINK
0150 26 29 017B          BNE     ZNK
                *
0152 B6 000E  D          LDAA    CU+2
0155 F6 000F  D          LDAB    CU+3
0158 BD 0000  P          JSR     RNGCAL    CU LOW
015B B7 0006  A          STAA    LOWECO+6
015E F7 0007  A          STAB    LOWECO+7
                *
0161 B6 000C  D          LDAA    CU
0164 F6 000D  D          LDAB    CU+1
0167 BD 0000  P          JSR     RNGCAL    CU HI
016A B7 0006  A          STAA    HIECO+6
016D F7 0007  A          STAB    HIECO+7
                *
0170 7D 0000  A          TST     F2        CALLED FROM QFIX?
0173 27 03 0178          BEQ     J2        NO:RETURN INIT
0175 7E 0000  A          JMP     RTNPT     YES:RETURN TO QFIX
```

```
005   FRTQ    ***  IP & FP INPUT ROUTINE  ***

                   *
0178 7E 0000  A J2      JMP     RENTR     RETURN TO INIT
                   *
                   *
017B B6 0012  D ZNK     LDAA    ZN+2
017E F6 0013  D         LDAB    ZN+3
0181 BD 0000  P         JSR     RNGCAL    ZN LOW
0184 B7 0000  A         STAA    LOWECO
0187 F7 0001  A         STAB    LOWECO+1
                   *
018A B6 0010  D         LDAA    ZN
018D F6 0011  D         LDAB    ZN+1
0190 BD 0000  P         JSR     RNGCAL    ZN HI
0193 B7 0000  A         STAA    HIECO
0196 F7 0001  A         STAB    HIECO+1
                   *
0199 7D 0000  A         TST     F2
019C 27 03 01A1         BEQ     J3
019E 7E 0000  A         JMP     RTNPT
                   *
01A1 7E 0000  A J3      JMP     RENTR     RETURN TO INIT
                   *
                   *
                   ************************************************
                   *  SCALE IP FOR USE BY 10 BIT D/A
                   ************************************************
                   *
01A4 B6 0018  D SKIPZN LDAA    IPHI
01A7 F6 0019  D         LDAB    IPLO      SCALE IP FOR D/A USE
01AA 31                 INS               (DEC. #) * 256/5000
01AB 31                 INS               RESULT IS THE # OF BITS
01AC 31                 INS               TO GIVE THE PROPER MICRO
01AD 31                 INS               AMP'S FROM THE D/A
01AE 31                 INS
01AF 37                 PSHB
01B0 36                 PSHA
01B1 86 01    A         LDAA    #$01
01B3 C6 00    A         LDAB    #$00
01B5 37                 PSHB
01B6 36                 PSHA              PUT MULTIPLIER ON STACK
01B7 34                 DES
                   *
01B8 BD 0000  A         JSR     MULT16
                   *
01BB B7 0000  A         STAA    DENDHU    SET UP FOR DIVISION
01BE F7 0000  A         STAB    DENDHL
01C1 EE 03    A         LDX     3,X
01C3 FF 0000  A         STX     DENDLU
                   *
01C6 86 13    A         LDAA    #$13      LOAD DIVISOR
01C8 C6 88    A         LDAB    #$88
01CA B7 0000  A         STAA    DSORU     5000 DEC.
01CD F7 0000  A         STAB    DSORL
                   *
01D0 BD 0000  A         JSR     DIV32
                   *
01D3 B6 0000  A         LDAA    DENDLU    16 BIT ANSWER
01D6 F6 0000  A         LDAB    DENDLL
```

```
006   FRTQ    ***  IP & FP INPUT ROUTINE  ***

01D9 B7 001A  D        STAA    IP10H     BIT VALUE
01DC F7 001B  D        STAB    IP10L
                *
01DF B6 0014  D        LDAA    ZERO1     ZERO VALUE
01E2 F6 0015  D        LDAB    ZERO1+1
                *
01E5 F0 001B  D        SUBB    IP10L
01E8 B2 001A  D        SBCA    IP10H     SUB BIT VALUE
                *
01EB F7 001B  D        STAB    IP10L     10 BIT D/A : OUTPUT
01EE B7 001A  D        STAA    IP10H     INITIAL POTETIAL VALUE
                *
                *
                ***************************************************
                **     SCALING FOR END POINT DETERMINATION
                **       (DIF * 10/3)
                ***************************************************
                *
01F1 B6 0016  D        LDAA    DIFHI
01F4 F6 0017  D        LDAB    DIFLO     SCALE DIF BY 10/3
01F7 31                INS
01F8 31                INS
01F9 31                INS
01FA 31                INS
01FB 31                INS               SET UP STACK
01FC 37                PSHB
01FD 36                PSHA              MULTIPLICAN TO STACK
01FE 86 00     A       LDAA    #$00
0200 C6 0A     A       LDAB    #$0A
0202 37                PSHB
0203 36                PSHA              MULTIPLIER TO STACK
0204 34                DES
                *
0205 BD 0000   A       JSR     MULT16
                *
0208 B7 0000   A       STAA    DENDHU
020B F7 0000   A       STAB    DENDHL    DIVIDEND
020E EE 03     A       LDX     3,X
0210 FF 0000   A       STX     DENDLU
                *
0213 86 00     A       LDAA    #$00
0215 C6 03     A       LDAB    #$03      DIVISOR
0217 B7 0000   A       STAA    DSORU
021A F7 0000   A       STAB    DSORL
                *
021D BD 0000   A       JSR     DIV32
                *
0220 B6 0000   A       LDAA    DENDLU    PUT ANSWER INTO A & B
0223 F6 0000   A       LDAB    DENDLL
                *
                *
                ***************************************************
                *      FORM COMPARISON VALUE FOR END OF SCAN
                **      SCALE DIF FOR 12 BITS
                **      (DIF * 0FFF/2710)
                *   (VOLT RANGE/1000 * 4095 BITS/10 VOLTS = # A/D
                ***************************************************
                *
```

```
007   FRTQ   ***  IP & FP INPUT ROUTINE  ***

                     *
 0226 31                     INS
 0227 31                     INS
 0228 31                     INS
 0229 31                     INS
 022A 31                     INS
 022B 37                     PSHB
 022C 36                     PSHA              MULTIPLICAN TO STACK
 022D 86 0F     A            LDAA   #$0F
 022F C6 FF     A            LDAB   #$FF
 0231 37                     PSHB
 0232 36                     PSHA              MULTIPLIER TO STACK
 0233 34                     DES
                     *
 0234 BD 0000   A            JSR    MULT16
                     *
 0237 B7 0000   A            STAA   DENDHU
 023A F7 0000   A            STAB   DENDHL     DIVIDEND
 023D EE 03     A            LDX    3,X
 023F FF 0000   A            STX    DENDLU
                     *
 0242 86 27     A            LDAA   #$27
 0244 C6 10     A            LDAB   #$10       DIVISOR
 0246 B7 0000   A            STAA   DSORU
 0249 F7 0000   A            STAB   DSORL
                     *
 024C BD 0000   A            JSR    DIV32
                     *
 024F B6 0000   A            LDAA   DENDLU     16 BIT ANSWER INTO A & B
 0252 F6 0000   A            LDAB   DENDLL
 0255 B7 001C   D            STAA   FPCMVU     STORE A/D COMPARISON VALUE
 0258 F7 001D   D            STAB   FPCMVL
                     *
 025B 7F 001E   D            CLR    FRSTQF
 025E BE 0000   D            LDS    STK        RESET STAK PNTR
                     *
 0261 7D 0000   A            TST    F2         CALLED FROM QFIX?
 0264 27 03 0269             BEQ    J1         NO
 0266 7E 0000   A            JMP    RTNPT      YES: RETURN TO QFIX
                     *
 0269 7E 0000   A  J1         JMP    RETINT
                     *
 026C CE 0000   A  ERROR      LDX    #ERRMES
 026F 86 01     A            LDAA   #1
 0271 BD 0000   A            JSR    PRINT1
 0274 CE 0000   A            LDX    #ASCBUF
 0277 86 20     A            LDAA   #$20
 0279 A7 00     A            STAA   X
 027B 7E 007A   P            JMP    FRSTQ1
                     *
                     *
                     *
                            XREF   KBIN,PRINT,PRINT1,BINHI,BINLO,MES30
                            XREF   MULT16,DENDHU,DENDHL,DENDLU,DENDLL
                            XREF   DIV32,ASCBUF,RETINT,LOWECO,HIECO
                            XREF   ZINK,DSORU,DSORL,MES31,ERRMES,RENTR
                            XREF   F2,RTNPT
                     *
```

```
008   FRTQ    ***  IP & FP INPUT ROUTINE  ***

                   *
                   *
                          XDEF    IPHI,IPLO,IP10H,IP10L,FPCMVU,FPCMVL
                          XDEF    FRSTQF,FRSTQ,RNGCAL,CD,PB,CU,ZN,RNG
                   *
 0000                     DSCT
                   *
 0000     0002   A STK    RMB     2
 0002     0002   A TP     RMB     2
 0004     0004   A CD     RMB     4
 0008     0004   A PB     RMB     4
 000C     0004   A CU     RMB     4
 0010     0004   A ZN     RMB     4
 0014     0002   A ZERO1  RMB     2
 0016     0001   A DIFHI  RMB     1
 0017     0001   A DIFLO  RMB     1
 0018     0001   A IPHI   RMB     1
 0019     0001   A IPLO   RMB     1
 001A     0001   A IP10H  RMB     1
 001B     0001   A IP10L  RMB     1
 001C     0001   A FPCMVU RMB     1
 001D     0001   A FPCMVL RMB     1
 001E     0001   A FRSTQF RMB     1
 001F     0002   A IP     RMB     2
 0021     0001   A IPHI1  RMB     1
 0022     0001   A IPLO1  RMB     1
                   *
          01FC   A ZERO    EQU     $01FC    ADJUSTED TO GIVE PROPER
                   *                    READ OUT ON PANEL METER
                   *                .1 UA (D/A OUTPUT)/ MV (PANEL READ
                   *
                   *
                          END
ERRORS 00000
```

```
001   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

                          NAM    KBIN     VER. 5   8-2-79    CIAVELL
                     *
                     *  FILE NAMES:  &KBIN (S) /KBIN (R)
                     *
                          OPT    REL
                     *
                          TTL    ***  KEY-BOARD INPUT SUBROUTINE  **
                     *
                     ************************************************
                     *
                     *    THIS SUBROUTINE INPUTS DATA FROM A 16 KEY
                     *    KEYBOARD AND DISPLAYS IT ON LED'S AND STORES
                     *    IT FOR USE BY OTHER ROUTINES.
                     *
                     ************************************************
                     *
                     *
                     *
0000                      BSCT
                     *
0000    0001   A BLANK   RMB    1
0001    0006   A ASCNUM  RMB    6
                     *
0000                      PSCT
                     *
0000    03     A TTBL    FCB    $03,$9F,$25,$0D,$99,$49,$41,$1F,$01
0001    9F     A
0002    25     A
0003    0D     A
0004    99     A
0005    49     A
0006    41     A
0007    1F     A
0008    01     A
0009    09     A         FCB    $09,$11,$C0,$63,$85,$61,$71
000A    11     A
000B    C0     A
000C    63     A
000D    85     A
000E    61     A
000F    71     A
                     *
                     *
                     *
0010 B7 0004   D KBIN    STAA   MAXDIG    STORE SCALE FACTOR
0013 7F 0012   D         CLR    BINLO
0016 7F 0001   D CLEAR   CLR    CHRCNT
0019 7F 0000   D         CLR    DECFLG
                     *
001C 7F 0003   D         CLR    BCDCNT
001F 7F 0002   D         CLR    NUMCNT
0022 7F 0011   D         CLR    BINHI
0025 CE FFFF   A         LDX    #$FFFF    LOAD DIGIT1&2 WI
0028 FF 000B   D         STX    DIGIT1    BLANKS
002B CE EFEF   A         LDX    #$EFEF    FILL DIGIT3-6 WI
002E FF 000D   D         STX    DIGIT3    UNDERLINES
0031 FF 000F   D         STX    DIGIT5
0034 CE 0000   A         LDX    #0
```

```
002   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

0037 FF 0013  D          STX    BCDHI     CLEAR BCD BUFFER
003A CE 2020  A          LDX    #$2020
003D DF 01    B          STX    ASCNUM    CLEAR ASCII BUFFER
003F DF 03    B          STX    ASCNUM+2
0041 DF 05    B          STX    ASCNUM+4
0043 7C 0012  D          INC    BINLO     TEST TO SEE IF WE
0046 28 08 0050          BVC    LIGHTS    HAVE RET FROM AN ERROR
0048 7F 0011  D          CLR    BINHI     IF SO,
004B 7F 0012  D          CLR    BINLO
004E 20 10 0060          BRA    BEGIN     GO TO BEGIN
                 *
0050 BD 0000  A LIGHTS   JSR    DISPLA    DISPLAY LED'S
0053 B6 0000  A          LDAA   P7AP      READ KEY-BOARD
0056 84 1F    A          ANDA   #$1F      MASK UPPER 3 BITS
0058 27 F6 0050          BEQ    LIGHTS    NO DATA-LOOP BACK
                 *
005A F6 0012  D          LDAB   BINLO     TEST TO SEE IF WE
005D 5C                  INCB             HAVE RET FROM AN ERROR
005E 29 B6 0016          BVS    CLEAR     IF SO, GO TO CLEAR
0060 7C 0001  D BEGIN    INC    CHRCNT    INCREMENT CHRCNT
0063 84 0F    A          ANDA   #@17      MASK OFF UPPER 4 BITS
0065 81 09    A          CMPA   #@11      A BETWEEN 0 &9?
0067 2F 1C 0085          BLE    NUMBER    IF SO, GO TO NUMBER
0069 81 0D    A          CMPA   #@15      ELSE, A=-?
006B 26 03 0070          BNE    JUMP1     IF SO,GO TO MINUS
006D 7E 0115  P          JMP    MINUS
0070 81 0E    A JUMP1    CMPA   #@16      ELSE, A=.?
0072 26 03 0077          BNE    JUMP2     IF SO,GO TO DECML
0074 7E 012F  P          JMP    DECML
0077 81 0B    A JUMP2    CMPA   #@13      ELSE, A=CHG?
0079 27 9B 0016          BEQ    CLEAR     IF SO, GO TO CLEAR
007B 81 0A    A          CMPA   #@12      ELSE, A=ENT?
007D 26 03 0082          BNE    JUMP3     IF SO, GO TO ENTER
007F 7E 0170  P          JMP    ENTER
0082 7E 01F6  P JUMP3    JMP    UGOOF
                 *
0085 7C 0002  D NUMBER   INC    NUMCNT    INCREMENT NUMBER COUNT
0088 7C 0003  D          INC    BCDCNT    INCREMENT BCD COUNT
008B C6 04    A          LDAB   #4        BCDCNT <=4
008D F1 0003  D          CMPB   BCDCNT
0090 2D 02 0094          BLT    UGOF1     IF NOT, GO TO UGOOF
0092 20 03 0097          BRA    JUMP5
0094 7E 01F6  P UGOF1    JMP    UGOOF
0097 B7 0006  D JUMP5    STAA   TADRL     STORE A IN TADRL
009A B6 0011  D          LDAA   BINHI     LOAD A WI ACTUAL # OF DIGI
                 *           BEFORE DECIMAL PT.
009D 84 7F    A          ANDA   #$7F      MASK OFF SIGN BIT
009F B1 0004  D          CMPA   MAXDIG    TOO MANY DIGITS ENTERED?
00A2 2F 03 00A7          BLE    OKNUM
00A4 7E 01F6  P          JMP    UGOOF
                 *
00A7 B6 0006  D OKNUM    LDAA   TADRL     ELSE, PUT CHAR IN A
00AA F6 0003  D          LDAB   BCDCNT    1ST NUMBER?
00AD C1 01    A          CMPB   #1
00AF 27 0B 00BC          BEQ    BCDST     IF SO, GO TO BCDST
00B1 C6 04    A          LDAB   #4        ELSE, SHIFT BCD BUFFER
00B3 78 0014  D BCDL     ASL    BCDLO     LEFT 4 BITS
00B6 79 0013  D          ROL    BCDHI
```

```
003   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

 00B9 5A                  DECB
 00BA 2E F7 00B3          BGT     BCDL
 00BC BB 0014  D BCDST    ADDA    BCDLO     STORE # IN 4 LEAST SIGN
 00BF B7 0014  D          STAA    BCDLO     BITS OF BCDLO
 00C2 B6 0002  D          LDAA    NUMCNT
 00C5 81 01    A          CMPA    #1
 00C7 27 1B 00E4          BEQ     LEDST     IF SO, GO TO LEDST
 00C9 CE 0010  D          LDX     #DIGIT6   IF NOT, CAL # TIMES
 00CC 80 02    A          SUBA    #2        TO SHIFT LED BUFFER
 00CE 27 04 00D4          BEQ     SECNUM
 00D0 09          LEDL1   DEX
 00D1 4A                  DECA
 00D2 26 FC 00D0          BNE     LEDL1     SHIFT LED BUFFER
 00D4 B6 0002  D SECNUM   LDAA    NUMCNT    UNTIL ALL NUMBERS
 00D7 4A                  DECA               HAVE BEEN LEFTSHIFTED
 00D8 E6 00    A LEDL2    LDAB    X         ONE LOCATION
 00DA 09                  DEX
 00DB E7 00    A          STAB    X
 00DD 4A                  DECA
 00DE 27 04 00E4          BEQ     LEDST
 00E0 08                  INX
 00E1 08                  INX
 00E2 20 F4 00D8          BRA     LEDL2
 00E4 C6 00    A LEDST    LDAB    #$00
 00E6 F7 0005  D          STAB    TADRU
 00E9 FE 0005  D          LDX     TADRU     STORE NEWEST CHAR
 00EC E6 00    A          LDAB    TBL,X     IN LED BUFFER
 00EE F7 0010  D          STAB    DIGIT6
 00F1 B6 0006  D          LDAA    TADRL     PUT CHAR IN A
 00F4 8B 30    A          ADDA    #@60      CONVERT IT TO ASCII
 00F6 CE 0021  B ASCST    LDX     #ASCNUM
 00F9 FF 0007  D          STX     TASCII
 00FC F6 0007  D          LDAB    TASCII
 00FF F7 0005  D          STAB    TADRU     STORE IN ASCII BUFFER
 0102 F6 0001  D          LDAB    CHRCNT
 0105 F7 0006  D          STAB    TADRL
 0108 FE 0005  D          LDX     TADRU
 010B A7 00    B          STAA    ASCNUM-1,X
 010D 86 01    A          LDAA    #1
 010F BD 0000  A          JSR     GENTIM    KEY IN DELAY
                  *
 0112 7E 0050  P          JMP     LIGHTS
                  *
 0115 F6 0001  D MINUS    LDAB    CHRCNT    IS - THE FIRST CHAR?
 0118 C1 01    A          CMPB    #1
 011A 26 02 011E          BNE     UGOF5     IF NOT, GOT TO UGOOF
 011C 20 03 0121          BRA     JUMPA
 011E 7E 01F6  P UGOF5    JMP     UGOOF
 0121 86 FD    A JUMPA    LDAA    #$FD      IF SO, PUT - IN DIGIT1
 0123 B7 000B  D          STAA    DIGIT1
 0126 86 80    A          LDAA    #$80      SET MSB OF BINHI
 0128 B7 0011  D          STAA    BINHI
 012B 86 2D    A          LDAA    #@55      SEND OUT ASCII MINUS TO
 012D 20 C7 00F6          BRA     ASCST     ASCII BUFFER
                  *
 012F 86 01    A DECML    LDAA    #1
 0131 B1 0001  D          CMPA    CHRCNT
 0134 26 05 013B          BNE     NF
```

```
004   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

0136 B7 0002  D         STAA    NUMCNT
0139 20 1C 0157         BRA     DECST
013B F6 0000  D NF      LDAB    DECFLG
013E 5D                 TSTB
013F 26 02 0143         BNE     UGOF4
0141 20 03 0146         BRA     JUMP8
0143 7E 01F6  P UGOF4   JMP     UGOOF
0146 F6 000B  D JUMP8   LDAB    DIGIT1
0149 C1 FD    A         CMPB    #$FD
014B 26 0A 0157         BNE     DECST
014D C6 02    A         LDAB    #2
014F F1 0001  D         CMPB    CHRCNT
0152 26 03 0157         BNE     DECST
0154 B7 0002  D         STAA    NUMCNT
0157 F6 0010  D DECST   LDAB    DIGITS
015A C4 FE    A         ANDB    #$FE
015C F7 0010  D         STAB    DIGITS
015F B6 0011  D         LDAA    BINHI
0162 BB 0003  D         ADDA    BCDCNT
0165 B7 0011  D         STAA    BINHI
0168 86 2E    A         LDAA    #$2E
016A 7C 0000  D         INC     DECFLG
016D 7E 00F6  P         JMP     ASCST
                *
0170 7D 0000  D ENTER   TST     DECFLG   IS DECFLG SET?
0173 26 03 0178         BNE     OK1      YES-GO TO OK1
0175 7E 01F6  P         JMP     UGOOF    NO-DEC.PT. NOT ENTERED-ERR
0178 B6 0000  A OK1     LDAA    FRSTQF   FLAG SET?
017B 27 53 01D0         BEQ     INTINP   NO GO TO INTEGER INPUT
                *
                *       DECIMAL INPUT SECTION
                *
017D F6 0011  D         LDAB    BINHI    # DIG BEFORE DEC. PT.
0180 C4 7F    A         ANDB    #$7F     MASK SIGN BIT
                *
0182 27 25 01A9         BEQ     ZERO1    IF NO DIG. GO TO ZERO1
                *
0184 B6 0003  D         LDAA    BCDCNT
0187 81 04    A         CMPA    #4       4 #'S ENTERED?
0189 27 52 01DD         BEQ     ASCPR    YES- GO TO ASCPR
                *
018B 81 03    A         CMPA    #3
018D 27 12 01A1         BEQ     ZERO2    3 #'S/PACK WITH 1-0
                *
018F 81 02    A         CMPA    #2
0191 27 12 01A5         BEQ     ZERO3    2 #'S/PACK WITH 2-0'S
                *
0193 86 0C    A         LDAA    #12      OTHERWISE PACK WIHT 3-0'S
0195 78 0014  D BCDPAK  ASL     BCDLO
0198 79 0013  D         ROL     BCDHI    DO 12 SHIFTS
019B 4A                 DECA
019C 26 F7 0195         BNE     BCDPAK
                *
019E 7E 01DD  P         JMP     ASCPR
                *
01A1 86 04    A ZERO2   LDAA    #4       PACK WITH 1-0
01A3 20 F0 0195         BRA     BCDPAK
                *
```

```
005   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

 01A5 86 08    A ZERO3  LDAA   #8        PACK WITH 2-0'S
 01A7 20 EC 0195        BRA    BCDPAK
                *
 01A9 F6 0003  D ZERO1  LDAB   BCDCNT    4 #S ENTERED?
 01AC C1 04    A        CMPB   #4        YES=ERROR
 01AE 27 02 01B2        BEQ    HOP
 01B0 20 03 01B5        BRA    HOP1
 01B2 7E 01F6  P HOP    JMP    UGOOF
                *
 01B5 C1 03    A HOP1   CMPB   #3
 01B7 27 24 01DD        BEQ    ASCPR
                *
 01B9 C1 02    A        CMPB   #2        PACK WITH 1-0
 01BB 26 05 01C2        BNE    ZERO4
 01BD 86 04    A        LDAA   #4
 01BF 7E 0195  P        JMP    BCDPAK
                *
 01C2 C1 01    A ZERO4  CMPB   #1        1 # ENTERED?
 01C4 26 02 01C8        BNE    HOP2      NO- GO TO UGOOF
 01C6 20 03 01CB        BRA    HOP3
                *
 01C8 7E 01F6  P HOP2   JMP    UGOOF
                *
 01CB 86 08    A HOP3   LDAA   #8        YES-PACK WITH 2-0'S
 01CD 7E 0195  P        JMP    BCDPAK
                *
                *
                *     INTEGER INPUT SECTION
                *
                *
 01D0 F6 0011  D INTINP LDAB   BINHI     # OF DIG BEFORE DEC PT
 01D3 C4 7F    A        ANDB   #$7F      MASK SIGN BIT
                *
 01D5 F1 0003  D        CMPB   BCDCNT    CMP # DIG BEFORE DEC PT
 01D8 27 03 01DD        BEQ    ASCPR     TO TOTAL # OF DIGITS,IF =
 01DA 7E 01F6  P        JMP    UGOOF
                *
 01DD BD 0000  A ASCPR  JSR    PRINT     PRINT ASCII BFFER
 01E0 B6 0013  D        LDAA   BCDHI
 01E3 F6 0014  D        LDAB   BCDLO
 01E6 BD 0000  A        JSR    BCDBIN    CONVERT BCD TO BINARY
 01E9 F7 0012  D        STAB   BINLO
 01EC F6 0011  D        LDAB   BINHI     STORE BINARY RESULT
 01EF C4 80    A        ANDB   #$80      SAVING MSB OF BINHI
 01F1 1B                ABA              IF SET
 01F2 B7 0011  D        STAA   BINHI
 01F5 39                RTS
                *
 01F6 CE 83FD  A UGOOF  LDX    #$83FD    WRITE OUT U-GOOF
 01F9 FF 000B  D        STX    DIGIT1    U-
 01FC CE 4103  A        LDX    #$4103
 01FF FF 000D  D        STX    DIGIT3    GO
 0202 CE 0371  A        LDX    #$0371
 0205 FF 000F  D        STX    DIGIT5    OF
 0208 86 7F    A        LDAA   #$7F
 020A B7 0012  D        STAA   BINLO
 020D 7E 0050  P        JMP    LIGHTS
                *
```

```
206   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

                 *
                         XREF    DISPLA,PRINT,FRSTQ,BCDBIN
                         XREF    TBL,FRSTQF,P7AP,GENTIM
                 *
                 *
                 *
                         XDEF    BINHI,BINLO,BCDHI,BCDLO,DIGIT0,DIGI
                         XDEF    DIGIT3,DIGIT4,DIGIT5,DIGIT6,ASCNUM,
                         XDEF    TADRU,TADRL,TTBL,KBIN
                 *
0000                     DSCT
                 *
 0000    0001   A DECFLG RMB     1
 0001    0001   A CHRCNT RMB     1
 0002    0001   A NUMCNT RMB     1
 0003    0001   A BCDCNT RMB     1
 0004    0001   A MAXDIG RMB     1
 0005    0001   A TADRU  RMB     1
 0006    0001   A TADRL  RMB     1
 0007    0002   A TASCII RMB     2
 0009    0001   A COUNT  RMB     1
 000A    0001   A DIGIT0 RMB     1
 000B    0001   A DIGIT1 RMB     1
 000C    0001   A DIGIT2 RMB     1
 000D    0001   A DIGIT3 RMB     1
 000E    0001   A DIGIT4 RMB     1
 000F    0001   A DIGIT5 RMB     1
 0010    0001   A DIGIT6 RMB     1
 0011    0001   A BINHI  RMB     1
 0012    0001   A BINLO  RMB     1
 0013    0001   A BCDHI  RMB     1
 0014    0001   A BCDLO  RMB     1
                 *
                 *
                         END

ERRORS 00000
```

```
201   PRNT   ***  MAIN PRINTING SUBROUTINE  ***

                          NAM    PRNT     VER 2    4-20-78    CLAVEL
                     *
                     *  FILE NAMES:  &PRNT (S) / PRNT (R)
                     *
                          OPT    REL
                     *
                          TTL    ***  MAIN PRINTING SUBROUTINE  ***
                     *
                     ****************************************************
                     *
                     *    THIS SUBROUTINE FORMATS THE INPUT & OUTPUT
                     *    PARAMETER MESSAGES AND DOES THE PRINTING.
                     *
                     ****************************************************
                     *
                     *
0000                      BSCT
                     *
0000     0028   A ASCBUF RMB    40
                     *
                     *
0000                      PSCT
                     *
0000 FF 0000   D PRINT   STX    TMESU    SAVE ADR OF MESSAGE
0003 CE 0000   B         LDX    #ASCBUF  PUT ADR OF BUF IN TADRU
0006 FF 0000   A         STX    TADRU
0009 B6 0000   A         LDAA   TADRL
000C D6 00     B         LDAB   ASCBUF   CALLED FROM KBIN?
000E C1 45     A         CMPB   #$45     IF SO, CLEAR 1ST FIVE LOC
0010 27 04 0016          BEQ    FIVE
0012 C6 28     A         LDAB   #$28
0014 20 02 0018          BRA    SETLIM   IF NOT, CLEAR ENTIRE 40
0016 C6 05     A FIVE    LDAB   #5       WORD BUFFER
0018 1B          SETLIM  ABA
0019 B7 0000   A         STAA   TADRL
001C 86 20     A         LDAA   #$20     FILL BUFFER WI BLANKS
001E A7 00     A BLANKS  STAA   X
0020 08                  INX
0021 BC 0000   A         CPX    TADRU
0024 26 F8 001E          BNE    BLANKS
0026 C1 05     A         CMPB   #5       CALLED FROM KBIN?
0028 27 46 0070          BEQ    REPRNT   IF SO, GO TO REPRNT
002A 86 06     A         LDAA   #6
002C CE 0000   B         LDX    #ASCBUF  FILL ASCII PRINT BUF
002F C6 45     A         LDAB   #$45     WI E
0031 E7 00     A         STAB   X
0033 08                  INX
0034 C6 4E     A         LDAB   #$4E     N
0036 E7 00     A         STAB   X
0038 08                  INX
0039 C6 54     A         LDAB   #$54     T
003B E7 00     A         STAB   X
003D 08                  INX
003E C6 45     A         LDAB   #$45     E
0040 E7 00     A         STAB   X
0042 08                  INX
0043 C6 52     A         LDAB   #$52     R
0045 E7 00     A         STAB   X
```

```
002   PRNT   ***  MAIN PRINTING SUBROUTINE  ***

0047 08                    INX
0048 08                    INX
0049 FF 0000  A            STX     TADRU
004C FE 0000  D FILBUF     LDX     TMESU     TRANSFER MESSAGE BUFF
004F E6 00    A            LDAB    X         TO PRINT BUFFER
0051 4C                    INCA              UNTIL COLON IS REACHED
0052 C1 3A    A            CMPB    #$3A
0054 27 15 006B            BEQ     COLON     COLON FOUND?
0056 FE 0000  A            LDX     TADRU     IF NO, CONTINUE
0059 E7 00    A            STAB    X
005B 7C 0000  A            INC     TADRL
005E 7C 0001  D            INC     TMESL
0061 7D 0001  D            TST     TMESL
0064 26 03 0069            BNE     SK10
0066 7C 0000  D            INC     TMESU
0069 20 E1 004C SK10       BRA     FILBUF
                           *
006B B7 0002  D COLON      STAA    COLLOC    SAVE COLON LOC
006E 20 30 00A0            BRA     WRITB     GO TO PRINT BUFFER
0070 B6 0002  D REPRNT     LDAA    COLLOC
0073 B7 0000  A            STAA    TADRL     POINT TO LOC OF COLON
0076 C6 3D    A            LDAB    #$3D      PUT IN = SIGN
0078 FE 0000  A            LDX     TADRU
007B E7 00    B            STAB    ASCBUF,X
007D 8B 02    A            ADDA    #2
007F B7 0000  A            STAA    TADRL
0082 CE 0000  A            LDX     #ASCNUM   GET ADR OF ASCII #
0085 FF 0000  D            STX     TMESU
0088 F6 0000  A            LDAB    CHRCNT
008B FE 0000  D XFER       LDX     TMESU     TRANSFER ASCII #
008E A6 00    A            LDAA    X         TO ASCBUF
                           *
0090 FE 0000  A            LDX     TADRU
0093 A7 00    B            STAA    ASCBUF,X
0095 5A                    DECB
0096 27 08 00A0            BEQ     WRITB
0098 7C 0001  D            INC     TMESL
009B 7C 0000  A            INC     TADRL
009E 20 EB 008B            BRA     XFER
00A0 CE 0000  B WRITB      LDX     #ASCBUF   PRINT OUT ENTIRE BUFF
00A3 DF 05    A            STX     $05
00A5 86 01    A            LDAA    #1
00A7 97 07    A            STAA    $07
00A9 BD B2C3  A            JSR     $B2C3
00AC BD B2A0  A            JSR     $B2A0
00AF 39                    RTS
                           *
                           *
                           XREF    TADRU,TADRL,CHRCNT,ASCNUM
                           *
                           *
                           XDEF    COLLOC,ASCBUF,PRINT
                           *
0000                       DSCT
                           *
                           *
0000     0001  A TMESU     RMB     1
0001     0001  A TMESL     RMB     1
```

```
003   PRNT   ***  MAIN PRINTING SUBROUTINE  ***

 0002    0001  A COLLOC RMB    1
                *
                       END
ERRORS 00000
```

```
001   DSPL   ***  LED DISPLAY SUBROUTINE  ***

                          NAM    DSPL     VER 2    12-14-77      CLA
                     *
                     *  FILE NAMES:  &DSPL (S) / DSPL (R)
                     *
                          OPT    REL
                     *
                          TTL    ***  LED DISPLAY SUBROUTINE  ***
                     *
                     ****************************************************
                     *
                     *    SUBROUTINE TO DISPLAY DIGITS INPUT VIA THE
                     *    KEYBOARD ON THE SYSTEMS LED'S.  CALLED FROM
                     *     KBIN.
                     *
                     ****************************************************
                     *
                     *
0000 CE 0000   A DISPLA LDX    #DIGIT0  ADDR. OF 1ST DIG -1 INTO X
0003 7F 0000   D        CLR    DSPCNT   CLR DES.PT. COUNTER
0006 86 01     A        LDAA   #1
0008 B7 0000   A        STAA   PSBP     ENABLE 1ST LED
000B 5F                 CLRB
000C 53          DIGLOP COMB
000D F7 0000   A        STAB   PSAP     ENABLE ALL SEGMENTS
0010 F6 0000   D        LDAB   DSPCNT
0013 C1 06     A        CMPB   #6       6 DIGITS WRITTEN?
0015 26 01 0018         BNE    NOEXIT   NO- DON'T RETURN
0017 39                 RTS              OTHERWISE RETURN
                     *
0018 08          NOEXIT INX
0019 E6 00     A        LDAB   X
001B F7 0000   A        STAB   PSAP     PUT DIG ON LED
001E 5F                 CLRB
001F 5C          DELAY  INCB
0020 26 FD 001F         BNE    DELAY
0022 48                 ASLA
0023 B7 0000   A        STAA   PSBP     ENABLE NEXT LED
0026 7C 0000   D        INC    DSPCNT   INC COUNTER
0029 20 E1 000C         BRA    DIGLOP   LOOP TO BEGINNING
                     *
                     *
                     *
                          XREF   PSAP,PSBP,DIGIT0
                     *
                     *
                          XDEF   DISPLA
                     *
0000                      DSCT
                     *
0000     0001   A DSPCNT RMB    1
                     *
                          END
ERRORS 00000
```

SECTION III:
Main System Control Routines

```
001   RESET  ***  SYSTEM SET UP ROUTINE  ***

                         NAM    RESET    VER. 26   9-14-79  CLAVELL
                     *
                     *  FILE NAMES:  &REST (S) / REST (R) /
                     *
                         OPT    REL
                     *
                         TTL    ***  SYSTEM SET UP ROUTINE  ***
                     *
                     ************************************************
                     *
                     *    THIS ROUTINE INITIALIZES ALL THE PIA'S AND C
                     *    AND THEN SETS THE SYSTEM TO ITS INITIAL COND
                     *
                     ************************************************
                     *
                     *
                     *
0000                     ASCT
                     *
                     ***  ASSIGN PIA LOCATIONS  ***
                     *
          7000  A P1AP   EQU    $7000
          7001  A P1BP   EQU    $7001     A/D & MULTIPLEXER
          7002  A P1AC   EQU    $7002
          7003  A P1BC   EQU    $7003
                *
          7004  A P2AP   EQU    $7004     STATUS LED'S
          7005  A P2BP   EQU    $7005     & D/A
          7006  A P2AC   EQU    $7006
          7007  A P2BC   EQU    $7007
                *
          7008  A P3AP   EQU    $7008
          7009  A P3BP   EQU    $7009     STATUS LED DISPLAYS
          700A  A P3AC   EQU    $700A
          700B  A P3BC   EQU    $700B
                *
          700C  A P4AP   EQU    $700C
          700D  A P4BP   EQU    $700D     VALVES
          700E  A P4AC   EQU    $700E
          700F  A P4BC   EQU    $700F
                *
          7020  A P5AP   EQU    $7020
          7021  A P5BP   EQU    $7021     DEPTH & TEMP
          7022  A P5AC   EQU    $7022
          7023  A P5BC   EQU    $7023
                *
          7024  A P6AP   EQU    $7024
          7025  A P6BP   EQU    $7025     LED-CYCLE
          7026  A P6AC   EQU    $7026     OUTPUTS TO E.T. TIMER
          7027  A P6BC   EQU    $7027
                *
          7010  A P7AP   EQU    $7010
          7011  A P7BP   EQU    $7011     KEYBOARD INPUT
          7012  A P7AC   EQU    $7012     & CLOCK INPUT
          7013  A P7BC   EQU    $7013
                *
          701C  A P8AP   EQU    $701C
          701D  A P8BP   EQU    $701D     6-LED DISPLAY
```

```
002   RESET  ***  SYSTEM SET UP ROUTINE  ***

          701E  A P8AC   EQU    $701E
          701F  A P8BC   EQU    $701F
                *
 0000             DSCT
                *
 0000     0001  A YRH    RMB    1          LOC. FOR YEAR VALUE
 0001     0001  A YRL    RMB    1
 0002     0001  A DAYH   RMB    1          LOC. FOR DAY VALUE
 0003     0001  A DAYL   RMB    1
                *
                *     ::: PROGRAM SECTION  :::
                *
 0000             PSCT
                *
                *     ** RESET & INITIALIZE ALL PIA'S  **
                *     **  FLAGS AND COUNTERS  **
                *
 0000 8E 0F90  A START  LDS    #$0F90     INIT. THE STACK
                *
 0003 0F               SEI                SET INTERRUPT MASK
                *
 0004 C6 20    A       LDAB   #32
 0006 86 00    A       LDAA   #$00       INIT. ALL PIA'S
 0008 CE 7000  A       LDX    #P1AP
 000B A7 00    A LOOP1 STAA   X          FILL WITH 0'S
 000D 08               INX
 000E 5A               DECB
 000F 26 FA 000B       BNE    LOOP1
                *
 0011 86 F0    A       LDAA   #$F0       PIA 1 (PA0-7,PB0-3:INPUTS)
 0013 B7 7001  A       STAA   P1BP       (PB4-7:OUTPUTS)
                *
 0016 86 FF    A       LDAA   #$FF
 0018 B7 7004  A       STAA   P2AP       ALL OUTPUTS
 001B B7 7005  A       STAA   P2BP
                *
 001E B7 7008  A       STAA   P3AP       ALL OUTPUTS
 0021 B7 7009  A       STAA   P3BP
                *
 0024 B7 700C  A       STAA   P4AP       ALL OUTPUTS
 0027 B7 700D  A       STAA   P4BP
                *
 002A B7 7021  A       STAA   P5BP       PA0-7:INPUTS
                *                          PB0-7:OUTPUTS
                *
 002D B7 7024  A       STAA   P6AP       ALL OUTPUTS
 0030 B7 7025  A       STAA   P6BP
                *
                *                        PIA 7:  ALL INPUTS
                *
 0033 B7 701C  A       STAA   P8AP
 0036 B7 701D  A       STAA   P8BP       ALL OUTPUTS
                *
                *     ** SET UP CONTROL REGISTERS  **
                *
 0039 C6 34    A       LDAB   #$34
 003B 86 06    A       LDAA   #$06
                *
```

```
003   RESET  ***  SYSTEM SET UP ROUTINE  ***

 003D F7 7002  A        STAB   P1AC
 0040 B7 7003  A        STAA   P1BC
                *
                *
 0043 B7 7022  A        STAA   P5AC
                *
 0046 B7 7012  A        STAA   P7AC
 0049 F7 7013  A        STAB   P7BC
                *
 004C B7 701E  A        STAA   P8AC
 004F B7 701F  A        STAA   P8BC
                *
 0052 86 0D    A        LDAA   #$0D      CA1&CA2,CB1&CB2 SET UP
 0054 B7 700A  A        STAA   P3AC      FOR NEG. EDGE & UNMASKED
 0057 B7 700B  A        STAA   P3BC      THESE ARE FOR OVER-RIDE
                *                      FLAG INTERRUPTS
 005A B7 700E  A        STAA   P4AC
 005D B7 700F  A        STAA   P4BC
                *
 0060 B6 7008  A        LDAA   P3AP      DUMMY READS
 0063 B6 7009  A        LDAA   P3BP      TO CLR INTERRUPT
 0066 B6 700C  A        LDAA   P4AP      FLAGS
 0069 B6 700D  A        LDAA   P4BP
                *
 006C 86 06    A        LDAA   #%00000110 CA1 &CA2 MASKED
 006E B7 7006  A        STAA   P2AC
                *
 0071 86 07    A        LDAA   #$07      CB1 UNMASKED (OVR RESET)
 0073 B7 7007  A        STAA   P2BC      CB2 MASKED
                *
 0076 B6 7004  A        LDAA   P2AP      DUMMY READS
 0079 B6 7005  A        LDAA   P2BP
                *
 007C C6 3E    A        LDAB   #$3E
 007E F7 7026  A        STAB   P6AC      SET CONTROL REG
 0081 C6 36    A        LDAB   #$36      FOR E.T. CONTROL
 0083 F7 7027  A        STAB   P6BC
                *
 0086 86 1C    A        LDAA   #$1C      CB1 MASKED (GO INTERRUPT)
 0088 B7 7023  A        STAA   P5BC      CB2 UNMASKED (HALT INTERRU
                *
                *
 008B 86 60    A        LDAA   #%01100000 LIFT PEN & SCAN OFF
 008D B7 7005  A        STAA   P2BP
                *
 0090 86 25    A        LDAA   #%00100101 SET VALVES: V1A,V2A,V3B
 0092 B7 700C  A        STAA   P4AP
                *
 0095 86 03    A        LDAA   #3        1.5 SEC DELAY
 0097 BD 0000  A        JSR    GENTIM
                *
 009A 7F 700C  A        CLR    P4AP      RESET TIMER
 009D 86 A8    A        LDAA   #%10101000 SET VALVES: V7B,V8B,V6B
 009F B7 700D  A        STAA   P4BP
                *
 00A2 86 03    A        LDAA   #3
 00A4 BD 0000  A        JSR    GENTIM
                *
```

```
004   RESET  ***  SYSTEM SET UP ROUTINE  ***

 00A7 7F 703D  A          CLR     P4BP       RESET TIMER
                *
 00AA 86 C0    A          LDAA    #$C0       SET CYCLE LED
 00AC B7 7024  A          STAA    P6AP       TO 0
                *
 00AF 7F 7025  A          CLR     P6BP       CLR .E.T.
 00B2 7F 701D  A          CLR     P8BP       OFF LED SEG
                *
                *         ::: CLEAR ALL FLAGS  :::
                *
 00B5 7F 0000  A          CLR     POLFLG
 00B8 7F 0000  A          CLR     NUMBYH
 00BB 7F 0000  A          CLR     NUMBYL
 00BE 7F 0009  D          CLR     PRXFLG
 00C1 7F 000A  D          CLR     ACDFLG
 00C4 7F 0008  D          CLR     RCLFLG
 00C7 7F 000D  D          CLR     VAFLG
 00CA 7F 000E  D          CLR     VBFLG
 00CD 7F 000F  D          CLR     RESTFG
 00D0 7F 0010  D          CLR     TIMFLG
 00D3 7F 0012  D          CLR     CNTR2
 00D6 7F 0000  A          CLR     ASCBUF
 00D9 7F 0000  A          CLR     OVR
 00DC 7F 0000  A          CLR     FRSTQF     DEC./INTEGER FLAG
 00DF 7F 0000  A          CLR     SAMPLE     USED IN COMPT
 00E2 7F 0000  A          CLR     ERFLAG
                *
 00E5 0E                  CLI
                *
 00E6 86 0F    A          LDAA    #$0F
 00E8 B7 7025  A          STAA    P6BP       SET E.T. TO 1 SEC.
 00EB BD 0000  A          JSR     CT
 00EE C6 00    A          LDAB    #0
 00F0 F7 7025  A          STAB    P6BP
 00F3 BD 0000  A          JSR     CT         DUMMY DIGIT
 00F6 86 01    A          LDAA    #1         LSD = 1
 00F8 B7 7025  A          STAA    P6BP
 00FB BD 0000  A          JSR     CT
 00FE F7 7025  A          STAB    P6BP       SET OTHER DIGITS
 0101 BD 0000  A          JSR     CT         TO 0
 0104 BD 0000  A          JSR     CT
 0107 BD 0000  A          JSR     CT
 010A BD 0000  A          JSR     CT
 010D BD 0000  A          JSR     CT
                *
 0110 C6 28    A          LDAB    #40        BOOT ERROR MESG
 0112 CE 0000  A          LDX     #MESER     INTO RAM
 0115 FF 003B  D          STX     TMES
 0118 CE 0013  D          LDX     #MESERR
 011B FF 003D  D          STX     T1MES
                *
 011E FE 003B  D          LDX     TMES
 0121 A6 00    A BOOT     LDAA    0,X
 0123 FE 003D  D          LDX     T1MES
 0126 A7 00    A          STAA    0,X
 0128 08                  INX
 0129 FF 003D  D          STX     T1MES
 012C FE 003B  D          LDX     TMES
```

```
005   RESET  ***  SYSTEM SET UP ROUTINE  ***

 012F 08                    INX
 0130 FF 003B  D            STX     TMES
 0133 5A                    DECB
 0134 26 EB 0121            BNE     BOOT
                      *
 0136 7D 0000  A            TST     NUMBYL
 0139 26 16 0151            BNE     TABLE
                      *
 013B CE 0000  A            LDX     #TTBL1    BOOT IN 7-SEG TABLE
 013E FF 003B  D            STX     TMES
 0141 CE 0010  B            LDX     #TBL1
 0144 FF 003D  D            STX     T1MES
 0147 C6 0A    A            LDAB    #10
 0149 7C 0000  A            INC     NUMBYL
 014C FE 003B  D            LDX     TMES
 014F 20 D0 0121            BRA     BOOT
                      *
 0151 B6 0000  A TABLE      LDAA    NUMBYL
 0154 81 02    A            CMPA    #2
 0156 27 17 016F            BEQ     OUT
 0158 7C 0000  A            INC     NUMBYL
 015B C6 10    A            LDAB    #16       BOOT IN OTHER TABLE
 015D CE 0000  A            LDX     #TTBL
 0160 FF 003B  D            STX     TMES
 0163 CE 0000  B            LDX     #TBL
 0166 FF 003D  D            STX     T1MES
 0169 FE 003B  D            LDX     TMES
 016C 7E 0121  P            JMP     BOOT
                      *
 016F 7F 0000  A OUT        CLR     NUMBYL
                      *
 0172 CE 0000  A            LDX     #MES1     1 ST MESSAGE: YEAR?
 0175 BD 0000  A            JSR     PRINT
                      *
 0178 86 04    A            LDAA    #4        TAKE IN DATA
 017A BD 0000  A            JSR     KBIN      CONV, TO BIN
 017D B6 0000  A            LDAA    BINLO     & STORE
 0180 F6 0000  A            LDAB    BINHI
 0183 B7 0001  D            STAA    YRL
 0186 F7 0000  D            STAB    YRH
                      *
 0189 CE 0000  A            LDX     #MES2     ASK FOR JULIAN DATE
 018C BD 0000  A            JSR     PRINT
                      *
 018F 86 04    A            LDAA    #4
 0191 BD 0000  A            JSR     KBIN      INPUT
 0194 B6 0000  A            LDAA    BINLO
 0197 F6 0000  A            LDAB    BINHI
 019A B7 0004  D            STAA    DAYL0
 019D F7 0005  D            STAB    DAYH0
                      *
 01A0 BD 0000  A            JSR     SUB3      SPACE
 01A3 CE 0000  A            LDX     #MES3     ASK OP. TO SET CLOCK
 01A6 C6 02    A            LDAB    #2
 01A8 F7 0000  A            STAB    BLOCK
 01AB BD 0000  A            JSR     PRINT1
 01AE BD 0000  A            JSR     SUB3      SPACE
                      *
```

```
206   RESET  ***  SYSTEM SET UP ROUTINE  ***

01B1 BD 0000  A         JSR     TIME      DISPLAY TIME ON LED'S
                      *                       UNTIL GO PUSHED
                      *
                      *
                      *
01B4 7E 0000  A         JMP     INIT
                      **
                      *
                      *       *** CHECK IF "GO" BUTTON HAS BEEN PUSHED  *
                      *
01B7 B6 7021  A RETSET  LDAA    P5BP      DUMMY READ
01BA B6 7023  A GOLOOP  LDAA    P5BC      POLL GO BUTTON
01BD 2A FB 01BA         BPL     GOLOOP    NOT PUSHED-LOOP
                      *                       OTHERWISE JMP TO
01BF 7F 0000  A         CLR     INTFLG    MAIN CONTROL PROGM
01C2 7E 0000  A         JMP     PROGM
                      *
                      *
                      *
                              XREF    TIME,PRINT,MES1,MES2,MES3,PROGM
                              XREF    BINLO,BINHI,ASCBUF,GENTIM,PRINT1
                              XREF    NUMBYH,NUMBYL,POLFLG,OVR,CT,FRSTQF
                              XREF    SAMPLE,BLOCK,KBIN,ERFLAG
                              XREF    MESER,INTFLG,INIT,SUB3,TTBL,TTBL1
                      *
                      *
                      *
                              XDEF    P1AP,P1BP,P1AC,P1BC,P2AP,P2BP,P2AC
                              XDEF    P3AP,P3BP,P3AC,P3BC,P4AP,P4BP,P4AC
                              XDEF    P5AP,P5BP,P5AC,P5BC,P6AP,P6BP,P6AC
                              XDEF    P7AP,P7BP,P7AC,P7BC,P8AP,P8BP,P8AC
                              XDEF    YRH,YRL,DAYH,DAYL,EMPFLG
                              XDEF    TMFLG,PRXFLG,ACDFLG,RCLFLG,SCNFLG
                              XDEF    CNTR1,CNTR2,START,RETSET,MESERR
                              XDEF    P2BC,P4BC,P6BC,P8BC,TIMFLG
                              XDEF    TMES,T1MES,TBL,TBL1,DAYL0,DAYH0
                      *
                      *
0004                          DSCT
                      *
0004     0001  A DAYL3  RMB     1
0005     0001  A DAYH3  RMB     1
0006     0001  A EMPFL3 RMB     1
0007     0001  A TMFL3  RMB     1
0008     0001  A RCLFL3 RMB     1
0009     0001  A PRXFL3 RMB     1
000A     0001  A ACDFL3 RMB     1
000B     0001  A SCNFL3 RMB     1
000C     0001  A STDFL3 RMB     1
000D     0001  A VAFL3  RMB     1
000E     0001  A VBFL3  RMB     1
000F     0001  A RESTF3 RMB     1
0010     0001  A TIMFL3 RMB     1
0011     0001  A CNTR1  RMB     1
0012     0001  A CNTR2  RMB     1
0013     0028  A MESERR RMB     40
003B     0002  A TMES   RMB     2
003D     0002  A T1MES  RMB     2
```

```
007   RESET  ***  SYSTEM SET UP ROUTINE  ***

                       *
 0000                          BSCT
                       *
 0000     0010   A TBL        RMB     16
 0010     003A   A TBL1       RMB     10
                       *
                               END
ERRORS 00000
```

```
                         NAM    INIT     VER. 20 21  9-14-79 3-7-80  CLAVEL
                    *
                    *  FILE NAMES:   SINIT (S) / INIT (R) /
                    *
                         OPT    REL
                    *
                         TTL    ***  INITIALIZATION SUBROUTINE  ***
                    *
                    ****************************************************
                    *
                    *    THIS SUBROUTINE INITIALIZES THE INSTRUMENT P
                    *    AND INPUTS PARAMETER DATA.
                    *
                    ****************************************************
                    *
0000                     BSCT
                    *
0000      0002  A STCN1H RMB    2          ZN STD CONC. BUFFER
0002      0002  A STCN2H RMB    2          CD
0004      0002  A STCN3H RMB    2          PB
0006      0002  A STCN4H RMB    2          CU
                    *
0000                     PSCT
                    *
                    *
0000 7F 0011    D INIT   CLR    ZINK
0003 7F 0012    D        CLR    VPRECY
0006 7F 0000    A        CLR    F2
0009 7F 0003    D        CLR    INTFLG
000C 7C 0003    D        INC    INTFLG
                    *
000F CE 00DA    P        LDX    #A0        INITIALIZE JUMP TABLE
0012 FF 0008    B        STX    J0         IN BASE SECTION
0015 CE 00FC    P        LDX    #A1
0018 FF 000A    B        STX    J0+2
001B CE 011E    P        LDX    #A2
001E FF 000C    B        STX    J0+4
0021 CE 013A    P        LDX    #A3
0024 FF 000E    B        STX    J0+6
0027 CE 0156    P        LDX    #A4
002A FF 0010    B        STX    J0+8
002D CE 0178    P        LDX    #A5
0030 FF 0012    B        STX    J0+10
0033 CE 01A4    P        LDX    #A6
0036 FF 0014    B        STX    J0+12
0039 CE 01D4    P        LDX    #A7
003C FF 0016    B        STX    J0+14
003F CE 01F5    P        LDX    #A8
0042 FF 0018    B        STX    J0+16
0045 CE 0221    P        LDX    #A9
0048 FF 001A    B        STX    J0+18
004B CE 0243    P        LDX    #A10
004E FF 001C    B        STX    J0+20
0051 CE 025F    P        LDX    #A11
0054 FF 001E    B        STX    J0+22
0057 CE 027B    P        LDX    #A12
005A FF 0020    B        STX    J0+24
005D CE 02A6    P        LDX    #A13
```

```
002   INIT   ***  INITIALIZATION SUBROUTINE  ***

0060 FF 0022  B         STX    J0+26
0063 CE 02CC  P         LDX    #A14
0066 FF 0024  B         STX    J0+28
0069 CE 02F7  P         LDX    #A15
006C FF 0026  B         STX    J0+30
006F CE 031D  P         LDX    #A16
0072 FF 0028  B         STX    J0+32
0075 CE 0343  P         LDX    #A17
0078 FF 002A  B         STX    J0+34
007B CE 036D  P         LDX    #A18
007E FF 002C  B         STX    J0+36
0081 CE 0397  P         LDX    #A19
0084 FF 002E  B         STX    J0+38
0087 CE 03C1  P         LDX    #A20
008A FF 0030  B         STX    J0+40
008D CE 03F0  P         LDX    #A21
0090 FF 0032  B         STX    J0+42
0093 CE 041A  P         LDX    #A22
0096 FF 0034  B         STX    J0+44
0099 CE 043F  P         LDX    #A23
009C FF 0036  B         STX    J0+46
009F CE 0469  P         LDX    #A24
00A2 FF 0038  B         STX    J0+48
00A5 CE 0000  A         LDX    #FRSTQ
00A8 FF 003A  B         STX    J0+50
00AB CE 048E  P         LDX    #RENTR
00AE FF 003C  B         STX    J0+52
00B1 CE 04AB  P         LDX    #A27
00B4 FF 003E  B         STX    J0+54
00B7 CE 04CC  P         LDX    #A28
00BA FF 0040  B         STX    J0+56
                  *
                  *
00BD 7E 0000  A         JMP    FRSTQ    GET IP & FP
                  *
00C0 7F 0000  A RETINT  CLR    P4AP     RESET TIMER
                  *
                  *
                  *
00C3 CE 0000  A         LDX    #MES14
00C6 C6 04    A         LDAB   #4
00C8 F7 0000  A         STAB   BLOCK    SYSTEM INIT. MESG.
00CB BD 0000  A         JSR    SUB3
00CE BD 0000  A         JSR    SUB3
00D1 BD 0000  A         JSR    PRINT1
                  *
00D4 BD 0000  A         JSR    SUB3
00D7 BD 0000  A         JSR    SUB3
                  *
00DA CE 0000  A A2      LDX    #MES16   BEGIN ASKING
00DD BD 0000  A         JSR    PRINT    VP QUESTIONS
00E0 86 03    A         LDAA   #3       &
00E2 BD 0000  A         JSR    KBIN     GET INPUT VALUES
00E5 B6 0000  A         LDAA   BCDHI
00E8 F6 0000  A         LDAB   BCDLO
00EB B7 0006  D         STAA   VPHGH
00EE F7 0007  D         STAB   VPHGL    HG PLATTING TIME
00F1 BD 0000  A         JSR    SUB3
```

```
003   INIT   ***  INITIALIZATION SUBROUTINE  ***

                      *
00F4 7D 0000  A       TST     F2          CALLED FROM OFIX?
00F7 27 03 00FC       BEQ     A1          NO: CONT.
00F9 7E 0000  A       JMP     RTNPT       YES: RETURN TO OFIX
                      *
00FC CE 0000  A A1    LDX     #MES17
00FF BD 0000  A       JSR     PRINT
0102 86 03    A       LDAA    #4
0104 BD 0000  A       JSR     KBIN        SAMPLE TIME
0107 B6 0000  A       LDAA    BCDHI
010A F6 0000  A       LDAB    BCDLO
010D B7 0008  D       STAA    VPSAMH
0110 F7 0009  D       STAB    VPSAML
0113 BD 0000  A       JSR     SUB3
                      *
0116 7D 0000  A       TST     F2
0119 27 03 011E       BEQ     A2
011B 7E 0000  A       JMP     RTNPT
                      *
011E CE 0000  A A2    LDX     #MES18
0121 BD 0000  A       JSR     PRINT
0124 86 01    A       LDAA    #1          VALVE DELAY TIME
0126 BD 0000  A       JSR     KBIN
0129 B6 0000  A       LDAA    BCDLO
012C B7 000A  D       STAA    VPVS1
012F BD 0000  A       JSR     SUB3
                      *
0132 7D 0000  A       TST     F2
0135 27 03 013A       BEQ     A3
0137 7E 0000  A       JMP     RTNPT
                      *
013A CE 0000  A A3    LDX     #MES19
013D BD 0000  A       JSR     PRINT
0140 86 01    A       LDAA    #1          VALVE DELAY2 TIME
0142 BD 0000  A       JSR     KBIN
0145 B6 0000  A       LDAA    BCDLO
0148 B7 000B  D       STAA    VPVS2
014B BD 0000  A       JSR     SUB3
                      *
014E 7D 0000  A       TST     F2
0151 27 03 0156       BEQ     A4
0153 7E 0000  A       JMP     RTNPT
                      *
0156 CE 0000  A A4    LDX     #MES20
0159 BD 0000  A       JSR     PRINT
015C 86 03    A       LDAA    #3          SCAN TIME
015E BD 0000  A       JSR     KBIN
0161 B6 0000  A       LDAA    BCDHI
0164 F6 0000  A       LDAB    BCDLO
0167 B7 0004  D       STAA    SCANTH
016A F7 0005  D       STAB    SCANTL
016D BD 0000  A       JSR     SUB3
                      *
0170 7D 0000  A       TST     F2
0173 27 03 0178       BEQ     A5
0175 7E 0000  A       JMP     RTNPT
                      *
0178 CE 0000  A A5    LDX     #MES21
```

```
004   INIT   ***  INITIALIZATION SUBROUTINE  ***

 017B BD 0000  A         JSR     PRINT
 017E 86 03    A         LDAA    #3        FLUSHING TIME
 0180 BD 0000  A         JSR     KBIN
 0183 B6 0000  A         LDAA    BINLO
                 *
 0186 81 78    A         CMPA    #120
 0188 25 0B 0195         BCS     STA3
                 *
 018A CE 0000  A         LDX     #MES65
 018D BD 0000  A         JSR     PRINT1
 0190 BD 0000  A         JSR     SUB3
 0193 20 E3 0178         BRA     A5
                 *
 0195 48         STA3    ASLA              *2
 0196 B7 000C  D         STAA    VPFLSL
 0199 BD 0000  A         JSR     SUB3
                 *
 019C 7D 0000  A         TST     F2
 019F 27 03 01A4         BEQ     A6
 01A1 7E 0000  A         JMP     RTNPT
                 *
 01A4 CE 0000  A A6      LDX     #MES22
 01A7 BD 0000  A         JSR     PRINT
 01AA 86 03    A         LDAA    #3
 01AC BD 0000  A         JSR     KBIN      STD ADD TIME
 01AF B6 0000  A         LDAA    BINLO     IN BIN (0-255 SEC MAX)
                 *
 01B2 81 01    A         CMPA    #1        # MUST BE > 1 & < 120
 01B4 23 04 01BA         BLS     ER10
                 *
 01B6 81 78    A         CMPA    #120
 01B8 25 0B 01C5         BCS     STA1
                 *
 01BA CE 0000  A ER10    LDX     #MES65    ERROR MESG
 01BD BD 0000  A         JSR     PRINT1
 01C0 BD 0000  A         JSR     SUB3
 01C3 20 DF 01A4         BRA     A6
                 *
 01C5 48         STA1    ASLA              "A" * 2
 01C6 B7 000D  D         STAA    VPSTD
 01C9 BD 0000  A         JSR     SUB3
                 *
 01CC 7D 0000  A         TST     F2
 01CF 27 03 01D4         BEQ     A7
 01D1 7E 0000  A         JMP     RTNPT
                 *
 01D4 CE 0000  A A7      LDX     #MES40
 01D7 BD 0000  A         JSR     PRINT     MAKE ACID ADDITION?
 01DA 86 01    A         LDAA    #1
 01DC BD 0000  A         JSR     KBIN
 01DF B6 0000  A         LDAA    BINLO
 01E2 B7 0000  A         STAA    ACDFLG
 01E5 BD 0000  A         JSR     SUB3
                 *
 01E8 7D 0000  A         TST     F2
 01EB 27 03 01F0         BEQ     S1
 01ED 7E 0000  A         JMP     RTNPT
                 *
```

```
005   INIT   ***  INITIALIZATION SUBROUTINE  ***

01F0 7D 0000  A S1      TST     ACDFLG    MAKE ACID ADD.?
01F3 27 2C 0221         BEQ     A9        NO: DON'T ADD
                     *
01F5 CE 0000  A A8      LDX     #MES23
01F8 BD 0000  A         JSR     PRINT
01FB 86 03    A         LDAA    #3        ACID ADD TIME
01FD BD 0000  A         JSR     KBIN      INPUT BIN
0200 B6 0000  A         LDAA    BINLO     (0-255 SEC)
                     *
0203 81 78    A         CMPA    #120
0205 25 0B 0212         BCS     STA2
                     *
0207 CE 0000  A         LDX     #MES65
020A BD 0000  A         JSR     PRINT1
020D BD 0000  A         JSR     SUB3
0210 20 E3 01F5         BRA     A8
                     *
0212 48         STA2    ASLA              *2
0213 B7 000E  D         STAA    VPACID
0216 BD 0000  A         JSR     SUB3
                     *
0219 7D 0000  A         TST     F2
021C 27 03 0221         BEQ     A9
021E 7E 0000  A         JMP     RTNPT
                     *
0221 CE 0000  A A9      LDX     #MES24
0224 BD 0000  A         JSR     PRINT
0227 86 03    A         LDAA    #3        CO2 PURGE TIME
0229 BD 0000  A         JSR     KBIN
022C B6 0000  A         LDAA    BCDHI
022F F6 0000  A         LDAB    BCDLO
0232 B7 000F  D         STAA    VPPRGH
0235 F7 0010  D         STAB    VPPRGL
0238 BD 0000  A         JSR     SUB3
                     *
023B 7D 0000  A         TST     F2
023E 27 03 0243         BEQ     A10
0240 7E 0000  A         JMP     RTNPT
                     *
0243 CE 0000  A A10     LDX     #MES25
0246 BD 0000  A         JSR     PRINT
0249 86 01    A         LDAA    #1
024B BD 0000  A         JSR     KBIN      ZINK ANALYSIS?
024E B6 0000  A         LDAA    BINLO     YES=1,NO=0
0251 B7 0011  D         STAA    ZINK
0254 BD 0000  A         JSR     SUB3
                     *
0257 7D 0000  A         TST     F2
025A 27 03 025F         BEQ     A11
025C 7E 0000  A         JMP     RTNPT
                     *
025F CE 0000  A A11     LDX     #MES26
0262 BD 0000  A         JSR     PRINT
0265 86 01    A         LDAA    #1        RECIRCULATE MODE?
0267 BD 0000  A         JSR     KBIN      YES=1,NO=0
026A B6 0000  A         LDAA    BINLO
026D B7 0000  A         STAA    RCLFLG
0270 BD 0000  A         JSR     SUB3
```

```
006   INIT   ***  INITIALIZATION SUBROUTINE  ***

                      *
0273 7D 0000  A       TST     F2
0276 27 03 027B       BEQ     A12
0278 7E 0000  A       JMP     RTNPT
                      *
027B CE 0000  A A12   LDX     #MES27
027E BD 0000  A       JSR     PRINT
0281 86 01    A       LDAA    #1        STD ADD CYCLE #
0283 BD 0000  A       JSR     KBIN
0286 B6 0000  A       LDAA    BINLO
                      *
0289 81 01    A       CMPA    #1        MUST BE > 1
028B 2E 0B 0298       BGT     STA
                      *
028D CE 0000  A       LDX     #MES65    ERROR MESSG
0290 BD 0000  A       JSR     PRINT1
0293 BD 0000  A       JSR     SUB3
0296 20 E3 027B       BRA     A12
                      *
0298 B7 0012  D STA   STAA    VPRECY
029B BD 0000  A       JSR     SUB3
                      *
029E 7D 0000  A       TST     F2
02A1 27 03 02A8       BEQ     A13
02A3 7E 0000  A       JMP     RTNPT
                      *
02A6 7C 0000  A A13   INC     FRSTOF    DEC. INPUT
                      *
02A9 CE 0000  A       LDX     #MES7
02AC BD 0000  A       JSR     PRINT
02AF 86 03    A       LDAA    #3        CD STD CONC.
02B1 BD 0000  A       JSR     KBIN
02B4 B6 0000  A       LDAA    BINHI
02B7 F6 0000  A       LDAB    BINLO
02BA 97 02    B       STAA    STCN2H
02BC D7 03    B       STAB    STCN2H+1
02BE BD 0000  A       JSR     SUB3
                      *
02C1 7D 0000  A       TST     F2
02C4 27 06 02CC       BEQ     A14
02C6 7F 0000  A       CLR     FRSTOF
02C9 7E 0000  A       JMP     RTNPT
                      *
02CC 7C 0000  A A14   INC     FRSTOF
02CF CE 0000  A       LDX     #MES9
02D2 BD 0000  A       JSR     PRINT
02D5 86 03    A       LDAA    #3        PB STD CONC.
02D7 BD 0000  A       JSR     KBIN
02DA B6 0000  A       LDAA    BINHI
02DD F6 0000  A       LDAB    BINLO
02E0 97 04    B       STAA    STCN3H
02E2 D7 05    B       STAB    STCN3H+1
02E4 BD 0000  A       JSR     SUB3
                      *
02E7 7D 0000  A       TST     F2
02EA 27 0B 02F7       BEQ     A15
02EC 7F 0000  A       CLR     FRSTOF
02EF 7E 0000  A       JMP     RTNPT
```

```
007   INIT   ***  INITIALIZATION SUBROUTINE  ***

                      *
02F2 7D 0011  D       TST     ZINK
02F5 26 26 031D       BNE     A16
                      *
02F7 7C 0000  A A15   INC     FRSTQF
02FA CE 0000  A       LDX     #MES10
02FD BD 0000  A       JSR     PRINT
0300 86 03    A       LDAA    #3        CU STD
0302 BD 0000  A       JSR     KBIN
0305 B6 0000  A       LDAA    BINHI
0308 F6 0000  A       LDAB    BINLO
030B 97 06    B       STAA    STCN4H
030D D7 07    B       STAB    STCN4H+1
030F BD 0000  A       JSR     SUB3
                      *
0312 7D 0000  A       TST     F2
0315 27 2C 0343       BEQ     A17
0317 7F 0000  A       CLR     FRSTQF
031A 7E 0000  A       JMP     RTNPT
                      *
031D 7C 0000  A A16   INC     FRSTQF
0320 CE 0000  A       LDX     #MES11
0323 BD 0000  A       JSR     PRINT
0326 86 03    A       LDAA    #3        ZN STD
0328 BD 0000  A       JSR     KBIN
032B B6 0000  A       LDAA    BINHI
032E F6 0000  A       LDAB    BINLO
0331 97 00    B       STAA    STCN1H
0333 D7 01    B       STAB    STCN1H+1
0335 BD 0000  A       JSR     SUB3
                      *
0338 7D 0000  A       TST     F2
033B 27 05 0343       BEQ     A17
033D 7F 0000  A       CLR     FRSTQF
0340 7E 0000  A       JMP     RTNPT
                      *
                      *
0343 7C 0000  A A17   INC     FRSTQF
0346 CE 0000  A       LDX     #MES70    CD LOWER LIMIT
0349 BD 0000  A       JSR     PRINT
034C 86 01    A       LDAA    #1
034E BD 0000  A       JSR     KBIN
0351 B6 0000  A       LDAA    BINHI
0354 84 7F    A       ANDA    #$7F      MASK SIGN BIT
0356 F6 0000  A       LDAB    BINLO
0359 B7 0002  A       STAA    CD+2
035C F7 0003  A       STAB    CD+3
035F BD 0000  A       JSR     SUB3
                      *
0362 7D 0000  A       TST     F2
0365 27 06 036D       BEQ     A18
0367 7F 0000  A       CLR     FRSTQF
036A 7E 0000  A       JMP     RNG       RECALCULATE RANGES
                      *
036D 7C 0000  A A18   INC     FRSTQF
0370 CE 0000  A       LDX     #MES71    CD HIGH LIMIT
0373 BD 0000  A       JSR     PRINT
0376 86 01    A       LDAA    #1
```

```
008   INIT   ***  INITIALIZATION SUBROUTINE  ***

0378 BD 0000  A          JSR    KBIN
037B B6 0000  A          LDAA   BINHI
037E 84 7F    A          ANDA   #$7F
0380 F6 0000  A          LDAB   BINLO
0383 B7 0000  A          STAA   CD
0386 F7 0001  A          STAB   CD+1
0389 BD 0000  A          JSR    SUB3
                *
038C 7D 0000  A          TST    F2
038F 27 06 0397          BEQ    A19
0391 7F 0000  A          CLR    FRSTQF
0394 7E 0000  A          JMP    RNG
                *
0397 7C 0000  A A19      INC    FRSTQF
039A CE 0000  A          LDX    #MES72   PB LOWER LIMIT
039D BD 0000  A          JSR    PRINT
03A0 86 01    A          LDAA   #1
03A2 BD 0000  A          JSR    KBIN
03A5 B6 0000  A          LDAA   BINHI
03A8 84 7F    A          ANDA   #$7F
03AA F6 0000  A          LDAB   BINLO
03AD B7 0002  A          STAA   PB+2
03B0 F7 0003  A          STAB   PB+3
03B3 BD 0000  A          JSR    SUB3
                *
03B6 7D 0000  A          TST    F2
03B9 27 06 03C1          BEQ    A20
03BB 7F 0000  A          CLR    FRSTQF
03BE 7E 0000  A          JMP    RNG
                *
03C1 7C 0000  A A20      INC    FRSTQF
03C4 CE 0000  A          LDX    #MES73   PB UPPER LIMIT
03C7 BD 0000  A          JSR    PRINT
03CA 86 01    A          LDAA   #1
03CC BD 0000  A          JSR    KBIN
03CF B6 0000  A          LDAA   BINHI
03D2 84 7F    A          ANDA   #$7F
03D4 F6 0000  A          LDAB   BINLO
03D7 B7 0000  A          STAA   PB
03DA F7 0001  A          STAB   PB+1
03DD BD 0000  A          JSR    SUB3
                *
03E0 7D 0000  A          TST    F2
03E3 27 0B 03F0          BEQ    A21
03E5 7F 0000  A          CLR    FRSTQF
03E8 7E 0000  A          JMP    RNG
                *
03EB 7D 0011  D          TST    ZINK     ZINC ANALYSIS?
03EE 26 4F 043F          BNE    A23      YES: GO TO A23
                *
03F0 7C 0000  A A21      INC    FRSTQF
03F3 CE 0000  A          LDX    #MES74   CU LOWER LIMIT
03F6 BD 0000  A          JSR    PRINT
03F9 86 01    A          LDAA   #1
03FB BD 0000  A          JSR    KBIN
03FE B6 0000  A          LDAA   BINHI
0401 84 7F    A          ANDA   #$7F
0403 F6 0000  A          LDAB   BINLO
```

```
209   INIT   ***  INITIALIZATION SUBROUTINE  ***

 0406 B7 0002  A          STAA    CU+2
 0409 F7 0003  A          STAB    CU+3
 040C BD 0000  A          JSR     SUB3
                  *
 040F 7D 0000  A          TST     F2
 0412 27 06 041A          BEQ     A22
 0414 7F 0000  A          CLR     FRSTQF
 0417 7E 0000  A          JMP     RNG
                  *
 041A 7C 0000  A A22      INC     FRSTQF
 041D CE 0000  A          LDX     #MES75   CU UPPER LIMIT
 0420 BD 0000  A          JSR     PRINT
 0423 86 01    A          LDAA    #1
 0425 BD 0000  A          JSR     KBIN
 0428 B6 0000  A          LDAA    BINHI
 042B 84 7F    A          ANDA    #$7F
 042D F6 0000  A          LDAB    BINLO
 0430 B7 0000  A          STAA    CU
 0433 F7 0001  A          STAB    CU+1
 0436 BD 0000  A          JSR     SUB3
                  *
 0439 7F 0000  A          CLR     FRSTQF
 043C 7E 0000  A          JMP     RNG
                  *
 043F 7C 0000  A A23      INC     FRSTQF
 0442 CE 0000  A          LDX     #MES76   ZN LOWER LIMIT
 0445 BD 0000  A          JSR     PRINT
 0448 86 01    A          LDAA    #1
 044A BD 0000  A          JSR     KBIN
 044D B6 0000  A          LDAA    BINHI
 0450 84 7F    A          ANDA    #$7F
 0452 F6 0000  A          LDAB    BINLO
 0455 B7 0002  A          STAA    ZN+2
 0458 F7 0003  A          STAB    ZN+3
 045B BD 0000  A          JSR     SUB3
                  *
 045E 7D 0000  A          TST     F2
 0461 27 06 0469          BEQ     A24
 0463 7F 0000  A          CLR     FRSTQF
 0466 7E 0000  A          JMP     RNG
                  *
 0469 7C 0000  A A24      INC     FRSTQF
 046C CE 0000  A          LDX     #MES77   ZN UPPER LIMIT
 046F BD 0000  A          JSR     PRINT
 0472 86 01    A          LDAA    #1
 0474 BD 0000  A          JSR     KBIN
 0477 B6 0000  A          LDAA    BINHI
 047A 84 7F    A          ANDA    #$7F
 047C F6 0000  A          LDAB    BINLO
 047F B7 0000  A          STAA    ZN
 0482 F7 0001  A          STAB    ZN+1
 0485 BD 0000  A          JSR     SUB3
                  *
 0488 7F 0000  A          CLR     FRSTQF
 048B 7E 0000  A          JMP     RNG
                  *
                  *
 048E CE 0000  A RENTR    LDX     #MES78
```

```
210   INIT   ***  INITIALIZATION SUBROUTINE  ***

0491 BD 0000  A         JSR     PRINT
0494 86 03    A         LDAA    #3
0496 BD 0000  A         JSR     KBIN        FILLING TIME
0499 B6 0000  A         LDAA    BINLO       (0-255 SEC MAX)
049C 48                 ASLA                *2
049D B7 0002  D         STAA    RSVFIL
04A0 BD 0000  A         JSR     SUB3
                *
04A3 7D 0000  A         TST     F2          CHK IF CALLED FROM
04A6 27 03 04AB         BEQ     A27         OTHER ROUTINE.
04A8 7E 0000  A         JMP     RTNPT
                *
04AB CE 0000  A A27     LDX     #MES80
04AE BD 0000  A         JSR     PRINT
04B1 86 01    A         LDAA    #1          WANT MED EXCHG?
04B3 BD 0000  A         JSR     KBIN
04B6 B6 0000  A         LDAA    BINLO
04B9 B7 0000  D         STAA    MEDEX
04BC BD 0000  A         JSR     SUB3
                *
04BF 7D 0000  A         TST     F2
04C2 27 03 04C7         BEQ     S2
04C4 7E 0000  A         JMP     RTNPT
                *
04C7 7D 0000  D S2      TST     MEDEX       MED EXCHG?
04CA 27 1D 04E9         BEQ     N1          NO: SKIP
                *
04CC CE 0000  A A28     LDX     #MES81      MED EXCHG FLUSHING
04CF BD 0000  A         JSR     PRINT       TIME (0-120 SEC)
04D2 86 02    A         LDAA    #2
04D4 BD 0000  A         JSR     KBIN
04D7 B6 0000  A         LDAA    BINLO
04DA 48                 ASLA                (*2)
04DB B7 0001  D         STAA    MEXCHG
04DE BD 0000  A         JSR     SUB3
                *
04E1 7D 0000  A         TST     F2
04E4 27 03 04E9         BEQ     N1
04E6 7E 0000  A         JMP     RTNPT
                *
04E9 CE 0000  A N1      LDX     #MES12
04EC C6 02    A         LDAB    #2          PRINT GO MESG
04EE F7 0000  A         STAB    BLOCK
04F1 BD 0000  A         JSR     SUB3
04F4 BD 0000  A         JSR     SUB3
04F7 BD 0000  A         JSR     PRINT1
04FA BD 0000  A         JSR     SUB3
04FD BD 0000  A         JSR     SUB3
                *
                *
0500 7E 0000  A         JMP     RETSET      RETURN TO RESET
                *
                        XREF    MES7,MES9,MES80,MES81
                        XREF    MES11,MES12,MES14,MES16,MES17
                        XREF    MES18,MES19,MES20,MES21,MES22,MES23
                        XREF    MES27,ACDFLG,MES40,MES10,MES24
                        XREF    MES25,MES26,MES78,SUB3,KBIN,FRSTQF
                        XREF    FRSTQ,BLOCK,PRINT1,PRINT
```

```
011   INIT    ***  INITIALIZATION SUBROUTINE  ***

                          XREF   BCDHI,BCDLO,BINHI,BINLO,P4AP,P2BP
                          XREF   P2BC,RETSET,RCLFLG,MES65,CU,PB,CD
                          XREF   ZN,MES70,MES71,MES72,MES73,MES74
                          XREF   MES75,MES76,MES77,RNG,F2,RTNPT
                     *
                     *
                          XDEF   SCANTH,SCANTL,STCN1H,STCN2H,RSVFIL
                          XDEF   STCN3H,STCN4H,VPHGH,VPHGL,MEDEX
                          XDEF   VPSAMH,VPSAML,VPVS1,VPVS2,MEXCHG
                          XDEF   VPFLSL,VPSTD,VPACID,VPPRGH,VPPRGL
                          XDEF   ZINK,VPRECY,INIT,RETINT,RENTR
                          XDEF   INTFLG,J0,A0,A1,A2,A3,A4,A5,A6,A27
                          XDEF   A7,A8,A9,A10,A11,A12,A13,A14,A15,A2
                          XDEF   A16,A17,A18,A19,A20,A21,A22,A23,A24
                     *
                     *
0000                      DSCT
                     *
0000     0001   A MEDEX   RMB    1
0001     0001   A MEXCHG  RMB    1
0002     0001   A RSVFIL  RMB    1
0003     0001   A INTFLG  RMB    1
0004     0001   A SCANTH  RMB    1
0005     0001   A SCANTL  RMB    1
0006     0001   A VPHGH   RMB    1
0007     0001   A VPHGL   RMB    1
0008     0001   A VPSAMH  RMB    1
0009     0001   A VPSAML  RMB    1
000A     0001   A VPVS1   RMB    1
000B     0001   A VPVS2   RMB    1
000C     0001   A VPFLSL  RMB    1
000D     0001   A VPSTD   RMB    1
000E     0001   A VPACID  RMB    1
000F     0001   A VPPRGH  RMB    1
0010     0001   A VPPRGL  RMB    1
0011     0001   A ZINK    RMB    1
0012     0001   A VPRECY  RMB    1
                     *
0008                      BSCT
                     *
0008     003A   A J0      RMB    58
                     *
                          END
ERRORS 00000
```

```
001   PROG    ***  SYSTEM CONTROL PROGRAM  ***

                         NAM    PROG     VER. 51  3-7-80   CLAVELL
                    *
                    *  FILE NAMES:   &PROG (S) / PROG (R) /
                    *
                         OPT    REL
                    *
                         TTL    ***  SYSTEM CONTROL PROGRAM  ***
                    *
                    ************************************************
                    *
                    * THIS IS THE MAIN CONTROL PROGRAM FOR THE ASV
                    * INSTRUMENT SYSTEM FOLLOWING SYSTEM INITIALIZATI
                    *
                    ************************************************
                    *
0000                     PSCT
                    *
0000   C0     A TTBL1  FCB    $C0,$79,$A4,$30,$99,$92,$03,$F8,$00
0001   79     A
0002   A4     A
0003   30     A
0004   99     A
0005   92     A
0006   03     A
0007   F8     A
0008   00     A
0009   18     A
                    *
                    *
0000                     DSCT
                    *
0000   07D0   A DATA   RMB    2000     STARTING ADDR. OF
                    *                   DATA BUFFER
                    *
000A                     PSCT
                    *
                    *    :::  MACRO DEFINITION  :::
                    *
                    ETM    MACR
                    *
                    \0 LDAB #$0F      LOAD COMAND
                     STAB P6BP
                     JSR CT
                     CLRA    DUMMY DIGIT
                     STAA P6BP
                     JSR CT
                     LDAA \1       LOAD BCD DIGITS
                     ANDA #$0F     MASK UPPER 4 BITS
                     STAA P6BP      STROBE IN LSD
                     JSR CT
                     LDAA \1       LOAD DIGITS AGAIN
                     LSRA         ROTATE RIGHT 4 TIMES
                     LSRA         TO GET 2 ND DIGIT
                     LSRA         INTO POSITION
                     LSRA
                     STAA P6BP   STROBE 2ND DIGIT
                     JSR CT
                     LDAA \2
```

```
002   PROG   ***  SYSTEM CONTROL PROGRAM  ***

                   ANDA #$0F
                   STAA P6BP   3RD DIGIT
                   JSR CT
                   LDAA \2
                   LSRA
                   LSRA
                   LSRA
                   LSRA
                   STAA P5BP  STROBE 4TH DIGIT
                   JSR CT
                   LDAA #$00   'PUT 0'S IN DIGITS
                   STAA P5BP    5 & 6
                   JSR CT
                   JSR CT
                   ENDM
                  *
                  ETMS   MACR
                  *
                   LDAA #$0F       LOAD COMAND
                   STAA P6BP
                   JSR CT
                   LDAB #$00        LOAD IN BCD DIGITS
                   STAB P6BP
                   JSR CT
                   JSR CT
                   LDAA \0
                   STAA P5BP
                   JSR CT
                   LDAA \1
                   STAA P5BP
                   JSR CT
                   STAB P6BP
                   JSR CT
                   JSR CT
                   JSR CT
                   ENDM
                  *
                  *
                  *
000A 7F 0000   A PROGM  CLR    CNTR1    CLEAR FLAGS
000D 7F 0000   A        CLR    SCNFLG
0010 7F 07DE   D        CLR    SEC
0013 7F 07DF   D        CLR    MIN
0016 7F 07E0   D        CLR    HR
0019 7F 0000   A        CLR    CSUMF
001C 7F 0000   A        CLR    HALTF
001F 7F 07DC   D        CLR    MIN0
0022 7F 07DD   D        CLR    HR0
0025 7F 07D1   D        CLR    PURGFG
0028 7F 07D0   D        CLR    DAYFG    DAY INC FLAG
                  *
002B B6 0030   A        LDAA   P6AP     DUMMY READ
                  *
002E CE 0030   A        LDX    #TIMBUF  INITIALIZE POINTER
0031 FF 0000   A        STX    TMBUF    USED IN COMPT
                  *
0034 CE 0000   D        LDX    #DATA
0037 FF 07E1   D        STX    EPDATA   CALCULATE NEXT TO
```

```
003   PROG   ***  SYSTEM CONTROL PROGRAM  ***

                      *                        LAST ADDR. IN DATA
003A F6 07E2  D       *       LDAB   EPDATA+1  BUFFER.  USED FOR
003D B6 07E1  D       *       LDAA   EPDATA    BUFFER OVER FLOW CHK.
                      *
0040 CB C8    A       *       ADDB   #$C8      ADD 1992 TO DATA
0042 89 07    A       *       ADCA   #$07
                      *
0044 B7 07E1  D       *       STAA   EPDATA
0047 F7 07E2  D       *       STAB   EPDATA+1
                      *
004A 7D 0000  A       *       TST    RCLFLG
004D 27 2C 007B       *       BEQ    LEDCT
                      *
004F BD 0000  A       *       JSR    CLOCK
                      *
0052 B6 0000  A       *       LDAA   TBCDS     GET TIME OF FIRST
0055 B7 07DB  D       *       STAA   SEC0      SAMPLE
0058 B6 0000  A       *       LDAA   TBCDM
005B B7 07DC  D       *       STAA   MIN0
005E B6 0000  A       *       LDAA   TBCDH
0061 B7 07DD  D       *       STAA   HR0
                      *
0064 7F 0000  A       *       CLR    P4AP
0067 7F 0000  A       *       CLR    P4BP
006A 86 10    A       *       LDAA   #$10      ACT V3A
006C B7 0000  A       *       STAA   P4AP
                      *
006F 86 04    A       *       LDAA   #4
0071 BD 0000  A       *       JSR    GENTIM
                      *
0074 86 60    A       *       LDAA   #$60      FILLING & SAMPLE
0076 B7 0000  A       *       STAA   P3AP      LED'S ON
0079 20 05 0080       *       BRA    FILL
                      *
007B 86 A0    A LEDCT *       LDAA   #$A0      FILLING & CONT.
007D B7 0000  A       *       STAA   P3AP      LED'S ON
                      *
0080 86 01    A FILL  *       LDAA   #$01
0082 B7 0000  A       *       STAA   P4BP      FILL PUMP ON
                      *
0085 B6 0000  A       *       LDAA   RSVFIL    FILLING TIME
0088 BD 0000  A       *       JSR    GENTIM
                      *
008B 7F 0000  A       *       CLR    P4BP      FILL PUMP OFF
008E B6 0000  A       *       LDAA   P3AP
0091 84 C0    A       *       ANDA   #$C0      FILLING LED OFF
0093 B7 0000  A       *       STAA   P3AP
                      *
                      *
0096 B6 0000  A       *       LDAA   ACDFLG    ADD ACID?
0099 27 26 00C1       *       BEQ    SKIP3     NO: SKIP
                      *
009B B6 0000  A       *       LDAA   P3AP      ACID PUMP ON
009E 8B 10    A       *       ADDA   #$10
00A0 B7 0000  A       *       STAA   P3AP
                      *
00A3 B6 0000  A       *       LDAA   P3BP      LED ON
00A6 8B 40    A       *       ADDA   #$40
```

```
004   PROG   ***  SYSTEM CONTROL PROGRAM  ***

00A8 B7 0000  A         STAA    P3BP
                *
00AB B6 0000  A         LDAA    VPACID    GET ADDITION TIME
00AE BD 0000  A         JSR     GENTIM    TIME IT
                *
00B1 B6 0000  A         LDAA    P3BP
00B4 84 BF    A         ANDA    #$BF      LED OFF
00B6 B7 0000  A         STAA    P3BP
                *
00B9 B6 0000  A         LDAA    P3AP
00BC 84 EF    A         ANDA    #$EF      ACID PUMP OFF
00BE B7 0000  A         STAA    P3AP
                *
00C1 C6 02    A SKIP3   LDAB    #2        INITIALIZE DATA
00C3 F7 0000  A         STAB    CNTR2     VALID FLAG
                *
00C6 7D 07D1  D RTN     TST     PURGFG    PURGE DONE?
00C9 26 FB 00C6         BNE     RTN       NO : LOOP
                *
00CB 7D 0000  A         TST     HALTF     HALT FLAG SET?
00CE 27 03 00D3         BEQ     RTN1      NO: CONT
00D0 BD 0000  A         JSR     HALT      YES: JUMP TO HALT ROUT
                *
00D3 7F 07D7  D RTN1    CLR     VBFLG     CLR STD & ACID FLAGS
00D6 7F 07D6  D         CLR     VAFLG
00D9 7F 0000  A         CLR     P4AP
00DC B6 0000  A         LDAA    P4BP      CLR VALVES
00DF 84 03    A         ANDA    #$03
00E1 B7 0000  A         STAA    P4BP
                *
00E4 CE 07D0  A         LDX     #2000
00E7 FF 07E3  D         STX     CNTHLD    CLEAR DATA BUFFER
00EA CE 0000  D         LDX     #DATA
00ED FF 07D2  D         STX     TDATA
                *
00F0 4F                 CLRA              FILL WITH 0'S
00F1 FE 07D2  D DCLR    LDX     TDATA
00F4 A7 00    A         STAA    0,X
00F6 08                 INX
00F7 FF 07D2  D         STX     TDATA
00FA FE 07E3  D         LDX     CNTHLD
00FD 09                 DEX
00FE FF 07E3  D         STX     CNTHLD
0101 26 EE 00F1         BNE     DCLR      DONE ?
                *
0103 7F 07D9  D         CLR     TADU
0106 F6 0000  A         LDAB    CNTR1
0109 F7 07DA  D         STAB    TADL      GET CYCLE NO.
                *
010C FE 07D9  D         LDX     TADU      CONVERT TO 7 SEG.
010F A6 00    A         LDAA    TBL1,X    DISPLY CODE
                *
0111 B7 0000  A         STAA    P6AP      SET CYCLE LED
                *
0114 7D 0000  A         TST     CNTR1
0117 27 1E 0137         BEQ     SKIPX
                *
0119 B6 07DB  D         LDAA    SEC0
```

```
005   PROG   ***  SYSTEM CONTROL PROGRAM  ***

011C B7 07DE  D        STAA   SEC
011F B6 07DC  D        LDAA   MIN0      GET CURRENT SAMPLE
0122 B7 07DF  D        STAA   MIN       TIME FOR USE
0125 B6 07DD  D        LDAA   HR0       BY COMP
0128 B7 07E0  D        STAA   HR
                  *
012B B6 0000  A        LDAA   DAYL0
012E B7 0000  A        STAA   DAYL
0131 B6 0000  A        LDAA   DAYH0
0134 B7 0000  A        STAA   DAYH
                  *
0137 86 64    A SKIPX  LDAA   #$64      START MAIN PUMP
                  *                      KEEP SCAN OFF
0139 B7 0000  A        STAA   P2BP       MET. PUMP OFF
013C 86 14    A        LDAA   #20        10 SEC DELAY LOOP
013E BD 0000  A        JSR    GENTIM
                  *
0141 CE 0000  D        LDX    #DATA      GET STARTING ADR.
0144 FF 07D2  D        STX    TDATA      DATA BUFF & STOR
                  *
0147 F6 0000  A        LDAB   IP10H      PUT IP UPPER INTO B
014A C4 03    A        ANDB   #$03       MASK UPPER 6 BITS
014C 86 E4    A        LDAA   #%11100100 LOAD A
014E 1B                ABA               ADD A&B INTO A
014F F6 0000  A        LDAB   IP10L      LOWER 8 BITS INTO B
0152 B7 0000  A        STAA   P2BP       SET D/A OUTPUT
0155 F7 0000  A        STAB   P2AP       ELECTRODES ON
                  *
0158 B6 0000  A        LDAA   P6AP       DUMMY READ/ZERO DETECT
                  *
015B                   ETM    SKIP4,VPHGL,VPHGH SET E.T.-HG PLT.
                  *
01A5 86 0A    A        LDAA   #$0A       SET HG LED
01A7 B7 0000  A        STAA   P3BP       ON=1
                  *
01AA B6 0000  A SKIP5  LDAA   P6AC       TIME UP?
01AD 2A FB 01AA        BPL    SKIP6      NO-LOOP
                  *
01AF B6 0000  A        LDAA   P6AP       DUMMY READ
                  *
01B2 B6 0000  A        LDAA   OVR        CK OVR
01B5 84 01    A        ANDA   #$01
01B7 26 05 01BE        BNE    SKIP8
                  *
01B9 86 02    A        LDAA   #$02       ACT V1B
01BB B7 0000  A        STAA   P4AP
                  *
01BE B6 0000  A SKIP8  LDAA   VPVS1      VP DELAY
01C1 48                ASLA              *2
01C2 BD 0000  A        JSR    GENTIM
                  *
01C5 B6 0000  A        LDAA   OVR        CK OVR
01C8 85 02    A        BITA   #$02
01CA 26 08 01D4        BNE    SKIP9
                  *                      OTHERWISE
01CC 7F 0000  A        CLR    P4AP       RESET
01CF 86 08    A        LDAA   #$08       & ACT V2B
01D1 B7 0000  A        STAA   P4AP
```

```
006   PROG   ***  SYSTEM CONTROL PROGRAM  ***

                     *
01D4 86 20     A SKIP9  LDAA   #$20      SAMPLE LED ON
01D6 B7 0000   A        STAA   P3BP      H3 & FULL LED'S OFF
                     *
                     *
01D9                    ETM    SKIP13,VPSAML,VPSAMH SAMP. TIME ON
                     *
                     *
0223 7D 0000   A        TST    RCLFLG    RECIRC MODE?
0226 27 0D 0235         BEQ    BRA6      NO:SKIP
                     *
0228 B6 0000   A        LDAA   P6AP      YES: DUMMY READ
022B B6 0000   A TIMUP  LDAA   P6AC      TIME UP ON E.T.?
022E 2A FB 022B         BPL    TIMUP     NO: LOOP
0230 B6 0000   A        LDAA   P6AP      DUMMY READ
                     *
0233 20 08 023D         BRA    SK12A
                     *
0235 B6 0000   A BRA6   LDAA   P1BP      DUMMY READ
                     *                       CLEAR RESV EMPTY FLG
0238 B6 0000   A SKIP12 LDAA   P1BC      CK FOR RESV EMPTY
023B 2A FB 0238         BPL    SKIP12    NO-LOOP
                     *
023D B6 0000   A SK12A  LDAA   P2BP
0240 84 FB     A        ANDA   #$FB
0242 B7 0000   A        STAA   P2BP      /PUMP OFF
                     *
0245 7F 0000   A        CLR    P3BP      SAMPLE PLATE LED OFF
                     *
0248 7D 0000   A        TST    MEDEX     USING MEDIUM EXCHG?
024B 27 5F 02AC         BEQ    SKIP7     NO: SKIP IT
                     *
                     ***********************************************
                     *      MEDIUM EXCHANGE
                     ***********************************************
                     *
024D B6 0000   A        LDAA   OVR
0250 85 08     A        BITA   #$08
0252 26 58 02AC         BNE    SKIP7     CK OVR
                     *
0254 7F 0000   A        CLR    P4AP      RESET TIMER
0257 7F 0000   A        CLR    P4BP
025A 86 04     A        LDAA   #$04      ACT V6A
025C B7 0000   A        STAA   P4BP
                     *
025F 86 02     A        LDAA   #2
0261 BD 0000   A        JSR    GENTIM
                     *
0264 7F 0000   A        CLR    P4BP
0267 86 01     A        LDAA   #$01      ACT V1A
0269 B7 0000   A        STAA   P4AP
                     *
026C 86 02     A        LDAA   #2
026E BD 0000   A        JSR    GENTIM
                     *
0271 7D 0000   A        TST    RCLFLG    V3 IN "A" POSIT?
0274 27 08 027E         BEQ    BRA3      NO:SKIP
                     *
```

```
007   PROG    ***  SYSTEM CONTROL PROGRAM  ***

0276 7F 0030  A         CLR     P4AP
0279 86 20    A         LDAA    #$20      ACT V3B
027B B7 0000  A         STAA    P4AP
                *
027E B6 0030  A BRA3    LDAA    P2BP
0281 8B 04    A         ADDA    #$04      PUMP ON
0283 B7 0030  A         STAA    P2BP
                *
0286 B6 0030  A         LDAA    MEXCHG    GET FLUSHING TIME
0289 BD 0000  A         JSR     GENTIM
                *
028C B6 0030  A         LDAA    P2BP
028F 84 FB    A         ANDA    #$FB      PUMP OFF
0291 B7 0030  A         STAA    P2BP
                *
0294 7F 0000  A         CLR     P4AP
0297 86 02    A         LDAA    #$02      ACT V1B
0299 B7 0000  A         STAA    P4AP
                *
029C 86 02    A         LDAA    #2
029E BD 0000  A         JSR     GENTIM
                *
02A1 7F 0030  A         CLR     P4BP
02A4 7F 0000  A         CLR     P4AP
02A7 86 08    A         LDAA    #$08      ACT V6B
02A9 B7 0000  A         STAA    P4BP
                *
02AC 86 1E    A SKIP7   LDAA    #30
02AE BD 0000  A         JSR     GENTIM    15 SEC EQUIL. TIME
                *
                *
                *
                ************************************************
                *           START SCAN
                ************************************************
                *
02B1 B6 0000  A SKIP13  LDAA    P2BP      PRESERVE 2 MSB
02B4 84 03    A         ANDA    #$03      OF D/A  /START CHART
02B6 C6 90    A         LDAB    #%10010000 PUT PEN DOWN
02B8 1B                 ABA               START SCAN & ELECTRODES ON
02B9 B7 0030  A         STAA    P2BP
02BC B6 0000  A         LDAA    P6AP      DUMMY READ
                *
02BF                    ETM     BLANK,SCANTL,SCANTH SET ET TO
                *                               SCAN TIME
0309 B6 0000  A         LDAA    CNTR2
030C 81 01    A         CMPA    #1        CNTR2 > 1
030E 2E 11 0321         BGT     SK16B     YES: SKIP
                *                          ELSE
0310 7F 0030  A         CLR     SCNFLG    CLR END OF SCAN FLG
0313 7C 07D8  D         INC     POLFLG    SET ENABLE SCAN FLAG
                *
                *
                ****    START TAKING DATA  ****
                *
0316 B6 0030  A         LDAA    P2AP      DUMMY READ
0319 86 07    A         LDAA    #07       CLR MASK ON CA1
031B B7 0000  A         STAA    P2AC      ENABLE INTERRUPTS FROM
```

```
008   PROG   ***  SYSTEM CONTROL PROGRAM  ***

031E 0E                    CLI              .2 SEC CLOCK INPUT
031F 20 05 0326            BRA    SKIP15
                    *
0321 86 07     A SK16B     LDAA   #$07      ENABLE ZERO INTERRUPT
0323 B7 0000   A           STAA   P6AC      ON E.T.
                    *
0326 7D 0000   A SKIP15    TST    RCLFLG
0329 27 0B 0336            BEQ    SK12
                    *
032B B6 0000   A           LDAA   CNTR1     LAST CYCLE?
032E B1 0000   A           CMPA   VPRECY
0331 27 03 0336            BEQ    SK12      YES: EMPTY RESV
                    *
0333 7E 03D6   P           JMP    SK1       NO: DON'T EMPTY
                    *
                    *
                    *
                    **********     FLUSH OUT RESV.     ************
                    *
0336 86 02     A SK12      LDAA   #$02
0338 B7 0000   A           STAA   P4BP      ACT V5B/START DRAINING
                    *
033B 86 10     A           LDAA   #$10      FLUSHING LED ON
033D B7 0000   A           STAA   P3BP
                    *
0340 7D 0000   A           TST    RCLFLG
0343 27 08 034D            BEQ    SK14A
                    *
0345 B6 0000   A           LDAA   P1BP      DUMMY READ
0348 B6 0000   A SK14C     LDAA   P1BC      RESV. EMPTY YET?
034B 2A FB 0348            BPL    SK14C     NO: LOOP
                    *                           YES: DO NORMAL FLUSH
                    *
034D 86 1E     A SK14A     LDAA   #30       15 SEC DELAY TO EMPTY
034F BD 0000   A           JSR    GENTIM    RESV. COMPLETELY
                    *
0352 C6 01     A           LDAB   #$01      STOP DRAINING
0354 F7 0000   A           STAB   P4BP      ACT V5A/FILL
                    *
0357 B6 0000   A           LDAA   OVR       CHK V5 OVR
035A 85 10     A           BITA   #$10      SKIP ALL FLUSHING
                    *                       TIMING IF V5
035C 27 03 0361            BEQ    SKIP17    IS IN OVERRIDE
                    *
035E 7E 03D6   P           JMP    SK1       NO FLUSH
                    *
                    *
0361 B6 0000   A SKIP17     LDAA   VPFLSL    FLUSH TIMING
0364 BD 0000   A           JSR    GENTIM    DO TIMING
                    *
                    *
0367 B6 0000   A SKIP20     LDAA   P1BP      DUMMY READ/CLR
                    *                       RESV. EMPTY FLAG
036A 86 02     A           LDAA   #$02      STOP FILLING
036C B7 0000   A           STAA   P4BP      ACT V5B/DRAIN
                    *
036F B6 0000   A SKIP22     LDAA   P1BC      RESV. EMPTY?
0372 2A FB 036F            BPL    SKIP22    YES-START FILLING
```

```
009   PROG   ***  SYSTEM CONTROL PROGRAM  ***

                   *                          OTHERWISE LOOP
                   *
0374 86 14     A SKIP23 LDAA   #20        10 SEC
0376 BD 0000   A        JSR    GENTIM     DELAY TO DRAIN COMPL
                   *
                   *
                   *
                   ************************************************
                   *****    START RESERVOIR FILLING      **********
                   ************************************************
                   *
0379 86 01     A        LDAA   #$01       STOP DRAINING
037B B7 0000   A        STAA   P4BP       ACT V5A/FILL
                   *
037E 7F 0000   A        CLR    P3BP       FLUSHING LED OFF
                   *
0381 B6 0000   A        LDAA   P3AP       FILLING LED ON
0384 8B 20     A        ADDA   #$20
0386 B7 0000   A        STAA   P3AP
                   *
0389 BD 0000   A        JSR    CLOCK
                   *
038C B6 0000   A        LDAA   TBCDS
038F B7 07DB   D        STAA   SEC0
0392 B6 0000   A        LDAA   TBCDM      GET TIME OF DAY
0395 B7 07DC   D        STAA   MIN0       SAMPLE COLLECTED
0398 B6 0000   A        LDAA   TBCDH      FOR USE BY COMPT ROUTINE
039B B7 07DD   D        STAA   HR0
                   *
039E 84 F0     A        ANDA   #$F0       MASK LOWER BYTE
03A0 81 02     A        CMPA   #2         UPPER BYTE = 2?
03A2 26 09 03AD         BNE    SKIP25     NO: HR0<23
03A4 B6 07DD   D        LDAA   HR0        RESET A
03A7 84 0F     A        ANDA   #$0F       MASK UPPER BYTE
03A9 80 03     A        SUBA   #3         A - 3
03AB 2C 15 03C2         BGE    SKIP26     IF>= 0 THEN HR0 = 23,24
                   *                            NO: HR0 < 23
03AD 7D 07D0   D SKIP25 TST    DAYFG      DAY FLAG SET?
03B0 27 13 03C5         BEQ    SKIP28     NO: CONT.
                   *
03B2 7F 07D0   D        CLR    DAYFG      OTHER WISE CLEAR FLAG
03B5 7C 0000   A        INC    DAYL0      INC DAYL0
03B8 7D 0000   A        TST    DAYL0      DAYL0 = 0?
03BB 26 08 03C5         BNE    SKIP28     NO: CONT.
                   *
03BD 7C 0000   A        INC    DAYH0      OTHER WISE INC DAYH0
03C0 20 03 03C5         BRA    SKIP28     CONT.
                   *
03C2 7C 07D0   D SKIP26 INC    DAYFG
                   *
03C5 B6 0000   A SKIP28 LDAA   RSVFIL     GET FILL TIME
03C8 BD 0000   A        JSR    GENTIM     RESV. FILLING TIME
                   *
03CB B6 0000   A        LDAA   P3AP
03CE 84 C0     A        ANDA   #$C0       FILLING LED OFF
03D0 B7 0000   A        STAA   P3AP
                   *
03D3 7F 0000   A        CLR    P4BP       FILL PUMP OFF
```

```
010   PROG   ***  SYSTEM CONTROL PROGRAM  ***

                 *
                 ************************************
                 *  SCANNING WAIT LOOP
                 ************************************
                 *
03D6 7D 0000  A SK1     TST     SCNFLG    SCAN DONE?
03D9 27 FB 03D5         BEQ     SK1       NO: LOOP
                 *
                 *
03DB B6 0000  A SKIP36  LDAA    P2BP      TURN MAIN PUMP
03DE 8B 04    A         ADDA    #%00000100 & PUMP LED ON
03E0 B7 0000  A         STAA    P2BP
                 *
03E3 B6 0000  A         LDAA    OVR
03E6 85 01    A         BITA    #$01
03E8 26 21 040B         BNE     SKIP37
                 *
03EA 7F 0000  A         CLR     P4AP
03ED B6 0000  A         LDAA    P4BP
03F0 84 03    A         ANDA    #$03
03F2 B7 0000  A         STAA    P4BP
03F5 86 01    A         LDAA    #$01      ACT V1A/
03F7 B7 0000  A         STAA    P4AP
                 *
03FA 7D 0000  A         TST     MEDEX     ADD 1 SEC EXTRA
03FD 27 05 0404         BEQ     VP1       DELAY IF DOING MED EX
                 *
03FF 86 02    A         LDAA    #2
0401 BD 0000  A         JSR     GENTIM
                 *
0404 B6 0000  A VP1     LDAA    VPVS2     GET VP DELAY 2
0407 48                 ASLA              *2
0408 BD 0000  A         JSR     GENTIM
                 *
040B B6 0000  A SKIP37  LDAA    OVR
040E 85 02    A         BITA    #$02
0410 26 33 0445         BNE     SKIP38
                 *
0412 B6 0000  A         LDAA    P4BP
0415 84 03    A         ANDA    #$03
0417 B7 0000  A         STAA    P4BP
041A 7F 0000  A         CLR     P4AP
041D 86 04    A         LDAA    #$04
041F B7 0000  A         STAA    P4AP      ACT V2A/HG LOOP
                 *
0422 86 02    A         LDAA    #2
0424 BD 0000  A         JSR     GENTIM
                 *
0427 7D 0000  A         TST     MEDEX
042A 27 19 0445         BEQ     SKIP38
                 *
042C 7D 0000  A         TST     RCLFLG    RECIRC MODE?
042F 27 14 0445         BEQ     SKIP38    NO: SKIP
                 *
0431 B6 0000  A         LDAA    OVR       YES: RESET V3
0434 85 04    A         BITA    #$04
0436 26 0D 0445         BNE     SKIP38
                 *
```

```
311   PROG   ***  SYSTEM CONTROL PROGRAM  ***

 0438 7F 0000  A         CLR    P4AP
 043B 86 10    A         LDAA   #$10      ACT V3A
 043D B7 0000  A         STAA   P4AP      RECIRC MODE
                  *
 0440 86 02    A         LDAA   #2
 0442 BD 0000  A         JSR    GENTIM
                  *
 0445 B6 0000  A SKIP38  LDAA   CNTR2     CNTR2=0?
 0448 27 03 044D         BEQ    SKIP39
 044A 7A 0000  A         DEC    CNTR2     NO : DEC CNTR2
                  *
                  *
 044D B6 0000  A SKIP39  LDAA   P6AP      DUMMY READ/CLEAR
                  *                      E.T. ZERO
 0450 7D 0000  A         TST    RCLFLG
 0453 27 0B 0460         BEQ    PURG
                  *
 0455 B6 0000  A         LDAA   CNTR1
 0458 B1 0000  A         CMPA   VPRECY
 045B 27 03 0460         BEQ    PURG      NEW SAMPLE- PURGE
 045D 7E 04BA  P         JMP    NOPURG    SAME SAMPLE - NO PURGE
                  *
                  *
 0460                    ETM    PURG,VPPRGL,VPPRGH PURGE TM-E.T.
                  *
 04AA B6 0000  A         LDAA   P3BP      PURGE LED ON
 04AD 8B 04    A         ADDA   #$04
 04AF B7 0000  A         STAA   P3BP
                  *
 04B2 86 07    A         LDAA   #$07      ENABLE E.T. 0
 04B4 B7 0000  A         STAA   P6AC      INTERRUPT
                  *
 04B7 7C 07D1  D         INC    PURGF3
                  *                      DUMMY READ TO
 04BA B6 0000  A NOPURG  LDAA   P1BP      CLR RESV. EMP. FLAG
                  *
 04BD 7D 0000  A         TST    RCLFLG
 04C0 27 0A 04CC         BEQ    SK25A
                  *
 04C2 B6 0000  A         LDAA   CNTR1
 04C5 B1 0000  A         CMPA   VPRECY    LAST CYCLE ?
 04C8 27 02 04CC         BEQ    SK25A     YES: SKIP
                  *
 04CA 20 2B 04F7         BRA    SKIP27    NO: NO ACID ADDITION
                  *
 04CC B6 0000  A SK25A   LDAA   ACDFLG    MAKE ACID ADD.?
 04CF 27 26 04F7         BEQ    SKIP27    NO-SKIP
                  *
 04D1 B6 0000  A         LDAA   P3AP      ACID PUMP ON
 04D4 8B 10    A         ADDA   #$10
 04D6 B7 0000  A         STAA   P3AP
                  *
 04D9 B6 0000  A         LDAA   P3BP
 04DC 8B 40    A         ADDA   #$40      ACID ADD LED ON
 04DE B7 0000  A         STAA   P3BP
                  *
 04E1 B6 0000  A         LDAA   VPACID    ACID ADD TIME
 04E4 BD 0000  A         JSR    GENTIM
```

```
012   PROG    ***  SYSTEM CONTROL PROGRAM  ***

                       *
04E7 B6 0000  A        LDAA   P3AP      ACID PUMP OFF
04EA 84 EF    A        ANDA   #$EF
04EC B7 0000  A        STAA   P3AP
                       *
04EF B6 0000  A        LDAA   P3BP
04F2 84 BF    A        ANDA   #$BF      ACID LED OFF
04F4 B7 0000  A        STAA   P3BP
                       *
                       *
04F7 B6 0000  A SKIP27 LDAA   CNTR1     TIME TO ADD STD?
04FA 8B 01    A        ADDA   #1        ADD 1 FOR LOOK AHEAD
04FC B1 0000  A        CMPA   VPRECY    VALUE AND COMPARE
04FF 26 26 0527        BNE    SKIP32    NO STD ADD-SKIP
                       *                    OTHERWISE CONT.
0501 B6 0000  A        LDAA   P2BP
0504 8B 08    A        ADDA   #$08      STD PUMP ON
0506 B7 0000  A        STAA   P2BP
                       *
0509 B6 0000  A        LDAA   P3BP
050C 8B 80    A        ADDA   #$80
050E B7 0000  A        STAA   P3BP      STD ADD LED ON
                       *
                       *
0511 B6 0000  A        LDAA   VPSTD     STD ADD TIME
0514 BD 0000  A        JSR    GENTIM
                       *
0517 B6 0000  A        LDAA   P2BP
051A 84 F7    A        ANDA   #$F7      STD PUMP OFF
051C B7 0000  A        STAA   P2BP
                       *
051F B6 0000  A        LDAA   P3BP
0522 84 7F    A        ANDA   #$7F      LED OFF
0524 B7 0000  A        STAA   P3BP
                       *
0527 BD 0000  A SKIP32 JSR    COMPT     COMPUTE ROUTINE
                       *
052A 7C 0000  A        INC    CNTR1
                       *
052D B6 0000  A        LDAA   VPRECY    GET STD ADD CYCLE #
0530 8B 01    A        ADDA   #1        INC BY 1
0532 B1 0000  A        CMPA   CNTR1     COMP TO CNTR1
0535 27 03 053A        BEQ    SKIP43    IF=,CLR CNTR1 & RTN
0537 7E 00C6  P        JMP    RTN
                       *
053A 7F 0000  A SKIP43 CLR    CNTR1
053D 7C 0000  A        INC    CNTR1
0540 7E 00C6  P        JMP    RTN
                       *
                       *
                       *
                       *
                              XREF   GENTIM,TBCDS,TBCDM,TBCDH,CNTR1
                              XREF   SCNFLG,CNTR2,MEDEX,MEXCHG
                              XREF   IP10L,VPVS1,VPSAML,VPSAMH,COMPT
                              XREF   SCAN,CLOCK,VPSTD,RSVFIL
                              XREF   BCDBIN,ACDFLG,VPRECY,DAYL,DAYH
                              XREF   VPPRGH,VPPRGL,P1AP,DAYL0,DAYH0
```

```
013   PROG    ***  SYSTEM CONTROL PROGRAM  ***

                                XREF    P1BP,P1AC,P1BC,P2AP,P2BP,P2AC,P2BC
                                XREF    P3AP,P3BP,P3AC,P3BC,P4AP,P4BP,P4AC
                                XREF    P4BC,P5AP,P6BP,P6AC,P6BC,HALT
                                XREF    VPHGH,VPHGL,IP10H,CSUMF,HALTF
                                XREF    RCLFLG,SCANTH,SCANTL,VPFLSL,CT,OVR
                                XREF    TIMBUF,TMBUF,VPACID,TBL1,VPVS2
                          *
                          *
                          *
                                XDEF    DATA,TDATA,PROGM,SEC,MIN,HR,CNTHLD
                                XDEF    VAFLG,VBFLG,POLFLG,EPDATA
                                XDEF    HR0,MIN0,SEC0,TTBL1,PURGFG
                          *
                          *
 07D0                           DSCT
                          *
 07D0     0001   A DAYFG   RMB     1
 07D1     0001   A PURGFG  RMB     1
 07D2     0002   A TDATA   RMB     2
 07D4     0001   A BCDMS   RMB     1
 07D5     0001   A BCDLS   RMB     1
 07D6     0001   A VAFLG   RMB     1
 07D7     0001   A VBFLG   RMB     1
 07D8     0001   A POLFLG  RMB     1
 07D9     0001   A TADU    RMB     1
 07DA     0001   A TADL    RMB     1
 07DB     0001   A SEC0    RMB     1
 07DC     0001   A MIN0    RMB     1
 07DD     0001   A HR0     RMB     1
 07DE     0001   A SEC     RMB     1
 07DF     0001   A MIN     RMB     1
 07E0     0001   A HR      RMB     1
 07E1     0002   A EPDATA  RMB     2
 07E3     0002   A CNTHLD  RMB     2
                   *
                   *
                                END
ERRORS 00000
```

```
 NAM MESG     VER. 17      9-19-79    CLAVELL
*
*   FILE NAMES:  &MESG (S) / MESG (R)
*
 OPT REL
*
 TTL    ***  MESSAGE LIST SUBROUTINE  ***
*
*****************************************************************
*
*   THIS IS A LIST OF ALL THE MESSAGES CALLED BY OTHER
*   SYSTEM PROGRAMES.
*
*****************************************************************
*
*
 PSCT

     FCC / YEAR:                                          /
     FCC / JULIAN DATE:                                   /
     FCC /SET REAL-TIME CLOCK TO CURRENT TIME:            /
     FCC /    THEN PUSH "GO"                              /

     FCC /13 CD STANDARD CONC.-IN PPB:                    /
     FCC /14 PB STANDARD CONC.-IN PPB:                    /
      FCC /15 CU STANDARD CONC.-IN PPB:                    /
      FCC /16 ZN STANDARD CONC.-IN PPB:                    /

      FCC / IF THE POTENTIOSTATE PARAMETERS               /
      FCC /   ARE SET, PUSH  "GO"                         /

      FCC /  * * * * * * * * * * * * * * * * * *         /
      FCC /  *    SYSTEM INITIALIZATION         *        /
      FCC /  *   ENTER THE FOLLOWING DATA       *        /
      FCC /  * * * * * * * * * * * * * * * * * *         /

      FCC /0 HG PLATTING TIME (MIN&SEC):                  /
      FCC /1 SAMPLE PLATE TIME(MIN&SEC):                  /
      FCC /2 VALVE DELAY1 (SEC):                          /
      FCC /3 VALVE DELAY2 (SEC):                          /
      FCC /4 SCAN TIME (MIN&SEC):                         /
      FCC /5 FLUSHING TIME (SEC):                         /
      FCC /6 STD. ADDITION TIME (SEC):                    /
      FCC /8 ACID ADDITION TIME (SEC):                    /
      FCC /9 PURGE DELAY TIME (MIN&SEC):                  /
      FCC /10 ZINC ANALYSIS (YES=1,NO=0):                 /
      FCC /11 RECIRC. MODE (YES=1,NO=0):                  /
      FCC /12 CYCLE # FOR STD. ADD .:                     /
      FCC /25 INITIAL POTENTIAL:                          /
      FCC /FINAL POTENTIAL:                               /
```

```
 FCC /INPUT ERROR: PLEASE REENTER NUMBER      /
FCC /***  SELECT PROGRAM MODE  ***            /
FCC /7 ACID ? (YES=1,NO=0):                   /
FCC /DATA OUTPUT  ** CONCENTRATIONS IN PPB **/
FCC /*** ERROR: LOW STD CONC-   ,  ,    ***  /
FCC /PARAMETER CHANGE (YES=1,NO=0):           /
FCC /PUSH "GO" TO RESTART PROGRAM             /
FCC /ERROR: NUMBER MUST BE > 1 & < 120        /
FCC /17 CD LOWER LIMIT (IN VOLTS):            /
FCC /18 CD UPPER LIMIT (IN VOLTS):            /
FCC /19 PB LOWER LIMIT (IN VOLTS):            /
FCC /20 PB UPPER LIMIT (IN VOLTS):            /
FCC /21 CU LOWER LIMIT (IN VOLTS):            /
FCC /22 CU UPPER LIMIT (IN VOLTS):            /
FCC /23 ZN LOWER LIMIT (IN VOLTS):            /
FCC /24 ZN UPPER LIMIT (IN VOLTS):            /
FCC /26 RESV FILLING TIME (SEC):              /
FCC /QUESTION NUMBER:                         /
FCC /27 MEDIUM EXCHG (YES=1,NO=0):            /
FCC /28 MEDIUM FLUSH TIME (SEC):              /


MES1,MES2,MES3,MES7,MES9,MES10,MES80,MES81
MES11,MES12,MES14,MES15,MES17,MES18,MES19
MES20,MES21,MES22,MES23,MES24,MES25,MES26,MES27
MES30,MES31,ERRMES,MES32,MES40,MES45,MESER
MES65,MES70,MES71,MES72,MES73,MES78
MES74,MES75,MES76,MES77,MES50,MES60,MES79
```

```
001   POLL   *** INTERRUPT POLLING ROUTINE ***

                          NAM     POLL      VER. 16  8-2-79  CLAVELL
                          OPT     REL
                      *
                      *  FILE NAMES:   &POL (S) / POL (R)
                      *
                      ****************************************************
                          TTL     *** INTERRUPT POLLING ROUTINE ***
                      ****************************************************
                      *
                      *
                      *
                      *******     INTERRRUPT CLOCK INTERVAL IS .2 SEC
                      *
0000                              PSCT
                      *
0000 B6 0000  A POLL    LDAA    POLFLG    SCANNING FLAG SET?
0003 27 08 000D         BEQ     POL       NO: GO TO POL
                      *
0005 B6 0000  A         LDAA    P2AC      SCAN INTERRUPT?
0008 2A 03 000D         BPL     POL       NO:  JMP TO POL
000A 7E 010C  P         JMP     POL3      YES: JMP TO SCAN
                      *
000D B6 0000  A POL     LDAA    P6AC      E.T. ZERO?
0010 2A 06 0018         BPL     HLT       NO: GO TO HLT
0012 7E 0152  P         JMP     POL17     YES:GO TO POL17
                      *
0015 7E 0124  P POLB    JMP     POL2
                      *
0018 B6 0000  A HLT     LDAA    P5BC      HALT FLAG SET?
001B 48                 ASLA
001C 2A 03 0021         BPL     VAL       NO: CHK OVR'S
001E 7E 013E  P         JMP     POL16     YES: JMP TO HALT
                      *
0021 B6 0000  A VAL     LDAA    P2BC      OVR-RESET SET?
0024 2B EF 0015         BMI     POLB
                      *
0026 C6 28    A         LDAB    #40
0028 CE 0000  A         LDX     #PRBUF    MOVE MESG INTO
002B FF 0000  A         STX     TMES      RAM PRINT BUFFER
002E CE 01AB  P         LDX     #MESOVR
0031 FF 0000  A         STX     T1MES
                      *
0034 FE 0000  A         LDX     T1MES
0037 A6 00    A BOOT1   LDAA    0,X
0039 FE 0000  A         LDX     TMES
003C A7 00    A         STAA    0,X
003E 08                 INX
003F FF 0000  A         STX     TMES      UPDATE POINTER
0042 FE 0000  A         LDX     T1MES
0045 08                 INX
0046 FF 0000  A         STX     T1MES     UPDATE POINTER
0049 5A                 DECB              DEC COUNTER
004A 26 EB 0037         BNE     BOOT1
                      *
004C B6 0000  A         LDAA    P3AC
004F 2B 1B 006C         BMI     POL4      V1?
                      *
0051 48                 ASLA
```

```
002   POLL   *** INTERRUPT POLLING ROUTINE ***

0052 2B 28 007C          BMI    POL5      V2?
                    *
0054 B6 0000  A          LDAA   P3BC
0057 2B 33 008C          BMI    POL6      V3?
                    *
0059 48                  ASLA
005A 2B 40 009C          BMI    POL7      V4?
                    *
005C B6 0000  A          LDAA   P4AC
005F 2B 4B 00AC          BMI    POL8      V5?
                    *
0061 48                  ASLA
0062 2B 58 00BC          BMI    POL9      V6?
                    *
0064 B6 0000  A          LDAA   P4BC
0067 2B 63 00CC          BMI    POL10     V7?
                    *
0069 48                  ASLA
006A 2B 70 00DC          BMI    POL11     V8?
                    *
                    *
006C B6 0001  D POL4     LDAA   OVR       TEST INTERRUPT REG.
006F 8B 01    A          ADDA   #$01
0071 B7 0001  D          STAA   OVR
0074 C6 31    A          LDAB   #$31
0076 F7 0006  A          STAB   PRBUF+5
0079 7E 00EC  P          JMP    BACK
                    *
007C B6 0001  D POL5     LDAA   OVR
007F 8B 02    A          ADDA   #$02
0081 B7 0001  D          STAA   OVR
0084 C6 32    A          LDAB   #$32
0086 F7 0006  A          STAB   PRBUF+6
0089 7E 00EC  P          JMP    BACK
                    *
008C B6 0001  D POL6     LDAA   OVR
008F 8B 04    A          ADDA   #$04
0091 B7 0001  D          STAA   OVR
0094 C6 33    A          LDAB   #$33
0096 F7 0006  A          STAB   PRBUF+6
0099 7E 00EC  P          JMP    BACK
                    *
009C B6 0001  D POL7     LDAA   OVR
009F 8B 08    A          ADDA   #$08
00A1 B7 0001  D          STAA   OVR
00A4 C6 34    A          LDAB   #$34
00A6 F7 0006  A          STAB   PRBUF+6
00A9 7E 00EC  P          JMP    BACK
                    *
00AC B6 0001  D POL8     LDAA   OVR
00AF 8B 10    A          ADDA   #$10
00B1 B7 0001  D          STAA   OVR
00B4 C6 35    A          LDAB   #$35
00B6 F7 0006  A          STAB   PRBUF+6
00B9 7E 00EC  P          JMP    BACK
                    *
00BC B6 0001  D POL9     LDAA   OVR
00BF 8B 20    A          ADDA   #$20
```

```
003   POLL   *** INTERRUPT POLLING ROUTINE ***

00C1 B7 0001  D         STAA   OVR
00C4 C6 36    A         LDAB   #$36
00C6 F7 0006  A         STAB   PRBUF+6
00C9 7E 00EC  P         JMP    BACK
                  *
00CC B6 0001  D POL10   LDAA   OVR
00CF 8B 40    A         ADDA   #$40
00D1 B7 0001  D         STAA   OVR
00D4 C6 37    A         LDAB   #$37
00D6 F7 0006  A         STAB   PRBUF+6
00D9 7E 00EC  P         JMP    BACK
                  *
00DC B6 0001  D POL11   LDAA   OVR
00DF 8B 80    A         ADDA   #$80
00E1 B7 0001  D         STAA   OVR
00E4 C6 38    A         LDAB   #$38
00E6 F7 0006  A         STAB   PRBUF+6
00E9 7E 00EC  P         JMP    BACK
                  *
                  *
00EC 86 01    A BACK    LDAA   #1       OVER RIDE MESS.
00EE 97 07    A         STAA   $0007
                  *
00F0 CE 0000  A         LDX    #PRBUF
00F3 DF 05    A         STX    $0005
                  *
00F5 BD B2C3  A         JSR    $B2C3
00F8 BD B2A0  A         JSR    $B2A0
00FB BD B38B  A         JSR    $B38B
                  *
00FE B6 0030  A         LDAA   P3AP     DUMMY READS TO
0101 B6 0020  A         LDAA   P3BP     CLEAR INTERRUPT
0104 B6 0010  A         LDAA   P4AP     FLAGS
0107 B6 0000  A         LDAA   P4BP
                  *
010A 0E                 CLI
010B 3B                 RTI
                  *
010C 7C 0000  D POL3    INC    CSUMF    ACCUMULATE 3 INTERRUPTS
010F B6 0000  D         LDAA   CSUMF    BEFORE COLLECTING
0112 81 03    A         CMPA   #3       DATA POINTS
0114 26 06 011C         BNE    RPO1     .6 SEC/PT
                  *
0116 7E 0000  A         JMP    SCAN
0119 7F 0000  D RPOL    CLR    CSUMF
                  *
011C B6 0030  A RPO1    LDAA   P2BP     DUMMY READ
011F B6 0010  A         LDAA   P2AP     DUMMY READ
                  *
0122 0E                 CLI
0123 3B                 RTI
                  *
0124 7F 0001  D POL2    CLR    OVR      CLR INTR. REGISTER
                  *
0127 86 01    A         LDAA   #1
0129 97 07    A         STAA   $0007
                  *
012B CE 0183  P         LDX    #MESP    POLLING MESG.
```

```
204   POLL    *** INTERRUPT POLLING ROUTINE ***

 012E DF 05     A          STX     $0005
                *
 0130 BD B2C3   A          JSR     $B2C3
 0133 BD B2A0   A          JSR     $B2A0
 0136 BD B38B   A          JSR     $B38B
                *
 0139 B6 0000   A          LDAA    P2BP       DUMMY READ
                *
 013C 0E                   CLI
 013D 3B                   RTI
                *
 013E 7D 0000   A POL15    TST     INTFLG     CALLED FROM INIT?
 0141 27 05 0148           BEQ     HLT1       NO
 0143 BD 0000   A          JSR     CHG        YES: JUMP TO HALT
                *
 0146 0E                   CLI
 0147 3B                   RTI
                *
 0148 86 01     A HLT1     LDAA    #1         SET THE HALT FLAG
 014A B7 0002   D          STAA    HALTF
 014D B6 0000   A          LDAA    P5BP       DUMMY READ
 0150 0E                   CLI
 0151 3B                   RTI
                *
 0152 86 06     A POL17    LDAA    #$06       MASK INTERRUPT
 0154 B7 0000   A          STAA    P6AC       ON E.T. ZERO
                *
 0157 7D 0000   A          TST     PURGFG     FROM PURGE?
 015A 27 0D 0169           BEQ     SCN        NO: SCAN
                *
 015C B6 0000   A          LDAA    P3BP
 015F 84 FB     A          ANDA    #$FB       PURGE LED OFF
 0161 B7 0000   A          STAA    P3BP
 0164 7F 0000   A          CLR     PURGFG
 0167 20 15 017E           BRA     SCN1
                *
 0169 7C 0000   A SCN      INC     SCNFLG     SET SCAN END FLAG
 016C 7F 0000   A          CLR     POLFLG
 016F 86 04     A          LDAA    #$04
 0171 B7 0020   A          STAA    P2AC
                *
 0174 B6 0000   A          LDAA    P2BP
 0177 84 03     A          ANDA    #$03       END SCAN
 0179 8B 60     A          ADDA    #$60
 017B B7 0000   A          STAA    P2BP
                *
 017E B6 0000   A SCN1     LDAA    P6AP       DUMMY READ
 0181 0E                   CLI
 0182 3B                   RTI
                *
                *
                          XREF    SCAN,P2AC,P2BC,P3AP,P3BP
                          XREF    P2BP,POLFLG,P2AP,P3AC,P3BC
                          XREF    P4AP,P4BP,P4AC,P4BC,P5BC
                          XREF    PRBUF,TMES,T1MES,PRNPG,P5BP
                          XREF    SCNFLG,P6AC,P6AP,PURGFG,INTFLG,CHG
                *
                          XDEF    POLL,RPOL,OVR,CSUMF,HALTF
```

```
005   POLL   *** INTERRUPT POLLING ROUTINE ***

                     *
0183    4F     A MESP   FCC     /OVER-RIDE FLAGS RESET
0184    56     A
0185    45     A
0186    52     A
0187    2D     A
0188    52     A
0189    49     A
018A    44     A
018B    45     A
018C    20     A
018D    46     A
018E    4C     A
018F    41     A
0190    47     A
0191    53     A
0192    20     A
0193    52     A
0194    45     A
0195    53     A
0196    45     A
0197    54     A
0198    20     A
0199    20     A
019A    20     A
019B    20     A
019C    20     A
019D    20     A
019E    20     A
019F    20     A
01A0    20     A
01A1    20     A
01A2    20     A
01A3    20     A
01A4    20     A
01A5    20     A
01A6    20     A
01A7    20     A
01A8    20     A
01A9    20     A
01AA    20     A
                     *
01AB    56     A MESOVR FCC     /VALVE   OVER-RIDE SET.
01AC    41     A
01AD    4C     A
01AE    56     A
01AF    45     A
01B0    20     A
01B1    20     A
01B2    20     A
01B3    4F     A
01B4    56     A
01B5    45     A
01B6    52     A
01B7    2D     A
01B8    52     A
01B9    49     A
01BA    44     A
```

```
006   POLL    *** INTERRUPT POLLING ROUTINE ***

 01BB     45     A
 01BC     20     A
 01BD     53     A
 01BE     45     A
 01BF     54     A
 01C0     2E     A
 01C1     20     A
 01C2     20     A
 01C3     20     A
 01C4     20     A
 01C5     20     A
 01C6     20     A
 01C7     20     A
 01C8     20     A
 01C9     20     A
 01CA     20     A
 01CB     20     A
 01CC     20     A
 01CD     20     A
 01CE     20     A
 01CF     20     A
 01D0     20     A
 01D1     20     A
 01D2     20     A
                 *
 0000                    DSCT
                 *
 0000     0001   A CSUMF  RMB     1
 0001     0001   A OVR    RMB     1
 0002     0001   A HALTF  RMB     1
                 *
                         END

ERRORS 00000
```

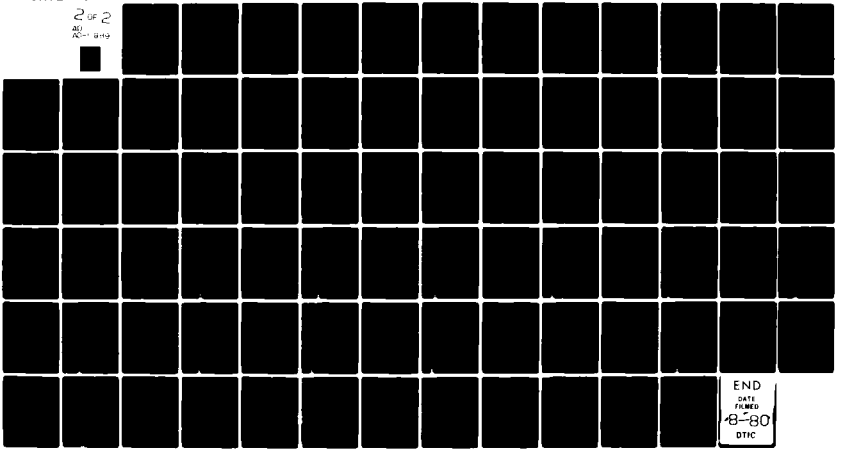
```
001   SCAN   *** DATA COLLECTING ROUTINE ***

                          NAM    SCAN     VER. 13   4-15-79    CLAVE
                      *
                      *     FILE NAMES:    &SCAN / SCAN (R)
                      *
                          OPT    REL
                      *
                          TTL    *** DATA COLLECTING ROUTINE ***
                      *
                      ****************************************
                      *
                      * ROUTINE TO COLLECT AND STORE "Y"
                      * DATA VALUES FOR USE BY COMPUT
                      * SUBROUTINE.  (1 X-Y VALUE PAIR / .6 SEC)
                      *
                      ****************************************
                      *
                      *
0000                      PSCT
                      *
0000 FE 0000  A SCAN  LDX    EPDATA   CHK NEXT TO LAST DATA BUFF
0003 A6 00    A       LDAA   X        ADDRESS FOR Ø'S,
0005 81 00    A       CMPA   #Ø       IF NOT Ø - BUFF OVER-
0007 27 03 000C       BEQ    OVRF     FLOW
0009 7E 008C  P       JMP    OVRFLO   GIVE OVER FLOW MESS
                      *
000C CE 000A  A OVRF  LDX    #10
000F FF 0000  D       STX    DTEN
                      *
0012 86 20    A       LDAA   #$20     Y AXIS/SET BIT 5
0014 B7 0000  A       STAA   P1BP
                      *
0017 86 3C    A       LDAA   #$3C     START A/D CONV.
0019 B7 0000  A       STAA   P1AC
001C 01               NOP
001D 01               NOP
001E 86 34    A       LDAA   #$34     RESET BIT
0020 B7 0000  A       STAA   P1AC
0023 B6 0000  A CONV  LDAA   P1AC     CHECK FOR CONV.
0026 2A FB 0023       BPL    CONV     COMPLETE-BIT 7 SET
                      *
0028 86 A7    A       LDAA   #167
002A 4A         D1    DECA             1 MS DELAY
002B 26 FD 002A       BNE    D1
                      *
002D B6 0000  A       LDAA   P1BP     GET UPPER BYTE OF Y
0030 84 0F    A       ANDA   #$0F     MASK UPPER 4 BITS
0032 F6 0000  A       LDAB   P1AP     GET LSB OF Y
                      *
                      ************************
                      *  DIVIDE Y VALUE BY 10
                      ************************
0035 CE 0000  D       LDX    #DTEN    GET ADDR OF HI
0038 BD 0000  A       JSR    DIV16    BYTE OF DIVISOR
                      *
003B FE 0000  A       LDX    TDATA    STORE IST Y VALUE
003E A7 00    A       STAA   X        MSB
0040 08               INX
0041 E7 00    A       STAB   X        LSB
```

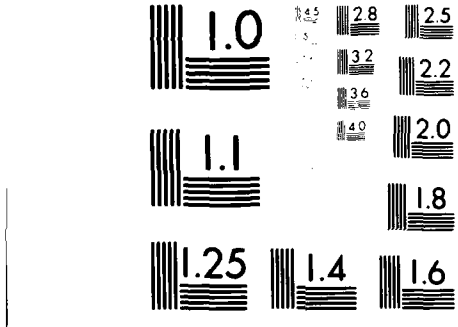
```
002   SCAN   *** DATA COLLECTING ROUTINE ***

0043 08                    INX             INCREMENT POINTER
0044 FF 0000  A            STX    TDATA    SAVE NEW ADDRESS
                  *
0047 FE 0002  D            LDX    NUMBYH   INC Y COUNTER
004A 08                    INX
004B FF 0002  D            STX    NUMBYH
                  *
004E 4F                    CLRA            X AXIS
004F B7 0000  A            STAA   P1BP
                  *
0052 86 3C    A            LDAA   #$3C     START A/D CONV.
0054 B7 0000  A            STAA   P1AC
0057 01                    NOP
0058 01                    NOP
0059 86 34    A            LDAA   #$34     RESET BIT
005B B7 0000  A            STAA   P1AC
005E B6 0000  A CONV1      LDAA   P1AC     CHECK FOR CONV.
0061 2A FB 005E            BPL    CONV1    COMPLETE-BIT 7 SET
                  *
0063 86 A7    A            LDAA   #167
0065 4A       D2           DECA
0066 26 FD 0065            BNE    D2
                  *
0068 B6 0000  A            LDAA   P1BP     GET MSB OF X
006B 84 0F    A            ANDA   #$0F     MASK UPPER 4 BITS
006D F6 0000  A            LDAB   P1AP     GET LSB OF X
                  *
0070 FE 0000  A            LDX    TDATA
0073 A7 00    A            STAA   X        STORE X VALUE
0075 08                    INX             TO DATA BUFF
0076 E7 00    A            STAB   X
0078 08                    INX
0079 FF 0000  A            STX    TDATA    INC BUFF POINTER
                  *
007C B1 0000  A            CMPA   FPCMVU   TEST FOR END PT.
007F 2A 03 0084            BPL    A2       IF=,  TEST LOW BYTE
                  *
0081 7E 0000  A            JMP    RPOL     OTHERWISE RETURN
                  *
0084 F1 0000  A A2         CMPB   FPCMVL   END PT. ?
0087 22 1C 00A5            BHI    CSCAN    YES, END SCAN
                  *
0089 7E 0000  A            JMP    RPOL
                  *
008C 86 04    A OVRFLO     LDAA   #$04     END SCAN
008E B7 0000  A            STAA   P2AC
0091 7C 0000  A            INC    SCNFLG   SCAN DONE IF SET
                  *
0094 CE 00BD  P            LDX    #MOVFLO  PRINT OUT OVER FLOW
0097 DF 05    A            STX    $0005    MESSAGE
                  *
0099 BD B2C3  A            JSR    $B2C3
009C BD B2A0  A            JSR    $B2A0
009F BD B38B  A            JSR    $B38B
                  *
00A2 7E 00AD  P            JMP    ENDSC    END SCAN
                  *
00A5 86 04    A CSCAN      LDAA   #$04     SCAN DONE
```

```
003   SCAN    *** DATA COLLECTING ROUTINE ***

00A7 B7 0000  A         STAA    P2AC      SET INTR. MASK/PIA
                *
00AA 7C 0000  A         INC     SCNFLG
                *
00AD 7F 0000  A ENDSC   CLR     POLFLG    RESET ENABLE SCAN FLG
00B0 B6 0000  A         LDAA    P2BP      KEEP 2 MSB OF D/A
00B3 84 03    A         ANDA    #$03
00B5 8B 60    A         ADDA    #%01100000 PEN UP,CHART OFF
00B7 B7 0000  A         STAA    P2BP      ELECT. OFF,SCAN OFF
                *
00BA 7E 0000  A         JMP     RPOL
                *
                        XREF    P1BP,P1AP,P1AC,TDATA,FPCMVU,FPCMVL
                        XREF    P2AC,SCNFLG,RPOL,EPDATA,DIV16
                        XREF    POLFLG,P2BP
                *
                *
                        XDEF    NUMBYH,NUMBYL,SCAN
                *
00BD    20      A MOVFLO FCC    / **  DATA BUFFER FULL  **
00BE    2A      A
00BF    2A      A
00C0    20      A
00C1    20      A
00C2    44      A
00C3    41      A
00C4    54      A
00C5    41      A
00C6    20      A
00C7    42      A
00C8    55      A
00C9    46      A
00CA    46      A
00CB    45      A
00CC    52      A
00CD    20      A
00CE    46      A
00CF    55      A
00D0    4C      A
00D1    4C      A
00D2    20      A
00D3    20      A
00D4    2A      A
00D5    2A      A
00D6    20      A
00D7    20      A
00D8    20      A
00D9    20      A
00DA    20      A
00DB    20      A
00DC    20      A
00DD    20      A
00DE    20      A
00DF    20      A
00E0    20      A
00E1    20      A
00E2    20      A
00E3    20      A
```

AD-A085 989 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 7/2
MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM.(U)
SEP 79 C CLAVELL
UNCLASSIFIED NOSC/TR-532 NL

2 OF 2
AD
A085 989





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
004   SCAN    *** DATA COLLECTING ROUTINE ***

 00E4    20     A
 00E5    20     A
                *
 0000                    DSCT
 0000    0002   A DTEN    RMB     2
 0002    0001   A NUMBYH  RMB     1
 0003    0001   A NUMBYL  RMB     1
                *
                *
                        END
ERRORS 00000
```

```
N
 NAM SCAN     VER. 14   5-10-79    CLAVELL
*
*    FILE NAMES:    &SC@N / SC@N (R)
*
 OPT REL
*
 TTL *** DATA COLLECTING ROUTINE ***
*
*****************************************
*
* ROUTINE TO COLLECT AND STORE "Y"
* DATA VALUES FOR USE BY COMPUT
* SUBROUTINE.  (1 X-Y VALUE PAIR / .6 SEC)
*
*****************************************
*
*
 PSCT
*
SCAN LDX EPDATA    CHK NEXT TO LAST DATA BUFF
 LDAA X              ADDRESS FOR 0'S,
 CMPA #0             IF NOT 0 - BUFF OVER-
 BEQ OVRF           FLOW
 JMP OVRFLO        GIVE OVER FLOW MESS
*
OVRF LDX #10
 STX DTEN
*
 LDAA #$20    Y AXIS/SET BIT 5
 STAA P1BP
*
 LDAA #$3C    START A/D CONV.
 STAA P1AC
 NOP
 NOP
 LDAA #$34     RESET BIT
 STAA P1AC
CONV LDAA P1AC      CHECK FOR CONV.
 BPL CONV         COMPLETE-BIT 7 SET
*
 LDAA #167
D1 DECA             1 MS DELAY
 BNE D1
*
 LDAA P1BP       GET UPPER BYTE OF Y
 ANDA #$0F     MASK UPPER 4 BITS
 LDAB P1AP     GET LSB OF Y
*
```

```
************************
*  DIVIDE Y VALUE BY 10
************************
 LDX #DTEN       GET ADDR OF HI
 JSR DIV16         BYTE OF DIVISOR
*
 LDX TDATA     STORE IST Y VALUE
 STAA X          MSB
 INX
 STAB X        LSB
 INX     INCREMENT POINTER
 STX TDATA      SAVE NEW ADDRESS
*
 LDX NUMBYH     INC Y COUNTER
 INX
 STX NUMBYH
*
 CLRA      X AXIS
 STAA P1BP
*
 LDAA #$3C     START A/D CONV.
 STAA P1AC
 NOP
 NOP
 LDAA #$34     RESET BIT
 STAA P1AC
CONV1 LDAA P1AC     CHECK FOR CONV.
 BPL CONV1        COMPLETE-BIT 7 SET
*
 LDAA #167
D2 DECA
 BNE D2
*
 LDAA P1BP  GET MSB OF X
 ANDA #$0F    MASK UPPER 4 BITS
 LDAB P1AP      GET LSB OF X
*
 LDX TDATA
 STAA X             STORE X VALUE
 INX                TO DATA BUFF
 STAB X
 INX
 STX TDATA           INC BUFF POINTER
*
 JSR OUT1
 JSR OUT
 CMPA FPCMVU     TEST FOR END PT.
 BEQ A2      IF=,  TEST LOW BYTE
*
 JMP RPOL       OTHERWISE RETURN
*
A2 CMPB FPCMVL    END PT. ?
 BCC CSCAN     YES, END SCAN
*
 JMP RPOL
*
CVRFLO LDAA #$04     END SCAN
 STAA P2AC
 INC SCNFLG            SCAN DONE IF SET
*
 LDX #MOVFLO           PRINT OUT OVER FLOW
 STX $0005             MESSAGE
*
 JSR $B2C3
 JSR $B2A0
 JSR $B38B
```

```
 JMP ENDSC       END SCAN
*
CSCAN LDAA #$04    SCAN DONE
 STAA P2AC      SET INTR. MASK/PIA
 LDAA #$43
 JSR $E1D1  OUTPUT "C"
*
 INC SCNFLG
*
ENDSC CLR POLFLG       RESET ENABLE SCAN FLG
 LDAA P2BP          KEEP 2 MSB OF D/A
 ANDA #$03
 ADDA #%01100000      PEN UP,CHART OFF
 STAA P2BP            ELECT. OFF,SCAN OFF
*
 JMP RPOL
*
OUT STAA SAVA
 STAB SAVB
 LDX #SAVA
 JSR $E0C8
 JSR CR1
 LDAA SAVA
 LDAB SAVB
 RTS
*
OUT1 LDX #FPCMVU
 JSR $E0C8
 JSR CR1
 RTS
*
CR1 LDX #CR
 JSR $E07E
 RTS
*
SAVA RMB 1
SAVB RMB 1
CR FCB $D,$A,4
*
 XREF P1BP,P1AP,P1AC,TDATA,FPCMVU,FPCMVL
 XREF P2AC,SCNFLG,RPOL,EPDATA,DIV16
 XREF POLFLG,P2BP
*
*
 XDEF NUMBYH,NUMBYL,SCAN
*
MOVFLO FCC / **  DATA BUFFER FULL  **
*
 DSCT
DTEN RMB 2
NUMBYH RMB 1
NUMBYL RMB 1
*
*
 END
```

```
001   HALT   *** PROGRAM HALT ROUTINE ***

                          NAM    HALT     VER.2   8-30-79   CLAVELL
                      *
                      *   FILE NAMES:  $HALT (S) / HALT (R)
                      *
                          OPT    REL
                      *
                          TTL    *** PROGRAM HALT ROUTINE ***
                      *
                      ************************************************
                      * THIS ROUTINE HALTS THE MAIN PROG AT THE START
                      * OF A NEW CYCLE AND ALLOWS UP TO 20 PARAMETERS
                      * TO BE CHANGED. PROG IS RESTARTED AT CYCLE 1
                      * SET S = NEXT QUESTION TO BE CHNGED
                      * GO = NO MORE CHANGES
                      *       ************************
                      * HALT MAY BE USED TO SIMPLY STOP EXECUTION FOR
                      * A WHILE.  PROG IS RESTARTED WITH "GO".
                      ************************************************
                      *
                      *
0000                      PSCT
                      *
0000 CE 0000   A HALT     LDX    #MES50   CHANGE MESS.
0003 BD 0000   A          JSR    PRINT
0006 86 01     A          LDAA   #1       1 # BEFORE DEC PT
0008 BD 0000   A          JSR    KBIN     READ KEYBOARD
000B B6 0000   A          LDAA   BINLO
000E 4D                   TSTA            A=1?
000F 26 03 0014           BNE    CHG      YES: CHANGE PARAM
0011 7E 006E   P          JMP    GOCHK    NO: WAIT FOR "GO"
                      *
0014 7F 0000   D CHG      CLR    TX       QUEST COUNTER POINTER
0017 7F 0001   D          CLR    TX+1
001A CE 0000   A CHG1     LDX    #MES79   GET # OF QUEST
001D BD 0000   A          JSR    PRINT    TO BE CHANGED
0020 86 02     A          LDAA   #2
0022 BD 0000   A          JSR    KBIN
                      *
0025 B6 0000   A          LDAA   BINLO
0028 FE 0000   D          LDX    TX
002B A7 00     B          STAA   QBUF,X   QUEST # INTO BUFFER
                      *
002D 7C 0001   D          INC    TX+1     INC POINTER
0030 B6 0001   D          LDAA   TX+1
0033 81 15     A          CMPA   #21      MAX OF 20 PARAM
0035 27 14 004B           BEQ    GONOW    CHANGES
                      *
0037 B6 0000   A          LDAA   P8BP     DUMMY READS FOR
003A B6 0000   A          LDAA   P5BP     SET S & GO
                      *
003D B6 0000   A QLOOP    LDAA   P8BC     "SET S" ?
0040 48                   ASLA
0041 2B D7 001A           BMI    CHG1     YES: NEXT QUEST
                      *
0043 B6 0000   A          LDAA   P5BC     "GO"?
0046 2A F5 003D           BPL    QLOOP    NO: KEEP POLLING
                      *                      YES:NO MORE CHANGES
                      *
```

```
002   HALT   *** PROGRAM HALT ROUTINE ***

0048 B6 0000  A         LDAA   P5BP      DUMMY READ
                *
004B 7A 0001  D GONOW   DEC    TX+1
004E FE 0000  D         LDX    TX        GET BUFF OFFSET
0051 A6 00    B         LDAA   QBUF,X
0053 BD 0000  A         JSR    QFIX      QUEST RETRIEVAL SUBR.
0056 7D 0001  D         TST    TX+1      DONE?
0059 26 F0 004B         BNE    GONOW     NO: LOOP BACK
                *
005B 7D 0000  A         TST    INTFLG    CALLED FROM INIT?
005E 26 2D 008D         BNE    GO2       YES: RTS
                *
0060 BD 0000  A         JSR    CLSABF    CLR SAMP BUF/COMP
0063 FE 0000  A         LDX    TIMBUF
0066 FF 0000  A         STX    TMBUF     REINITIALIZE TIME BUF
0069 86 01    A         LDAA   #1
006B B7 0000  A         STAA   CNTR1     RESET CNTR1
                *
006E CE 0000  A GOCHK   LDX    #MES60    PUSH "GO" MESSG.
0071 86 01    A         LDAA   #1
0073 B7 0000  A         STAA   BLOCK
0076 BD 0000  A         JSR    SUB3      LF
0079 BD 0000  A         JSR    PRINT1
007C BD 0000  A         JSR    SUB3
007F BD 0000  A         JSR    SUB3
                *
0082 7F 0000  A         CLR    HALTF     CLR HALT FLAG
0085 B6 0000  A         LDAA   P5BP      DUMMY READ
0088 B6 0000  A G01     LDAA   P5BC      "GO" PUSHED?
008B 2A FB 0088         BPL    G01       NO: LOOP
                *
008D 39           GO2   RTS
                *
                *
                      XREF   MES50,MES60
                      XREF   PRINT1,PRINT,KBIN,SUB3,P5BP,P5BC
                      XREF   BLOCK,BINLO,BINHI,BCDLO,BCDHI,QFIX
                      XREF   STCN1H,STCN2H,STCN3H,STCN4H,FRSTQF
                      XREF   HALTF,VPRECY,ZINK,TIMBUF,MES79
                      XREF   CLSABF,CNTR1,TMBUF,P8BC,P8BP,INTFLG
                *
                      XDEF   HALT,CHG,QBUF
                *
                *
0000                  DSCT
                *
0000     0002  A TX   RMB    2
                *
0000                  BSCT
                *
0000     0014  A QBUF RMB    20        QUEST # BUFF
                *
                *
                      END
ERRORS 00000
```

```
001   QFIX   *** QUESTION REPEAT SUBROUTINE ***

                              NAM    QFIX     VER.1   7-30-79  CLAVELL
                         *
                         *   FILE NAMES:   &QFIX (S) / QFIX (R)
                         *
                              OPT    REL
                         *
                              TTL    *** QUESTION REPEAT SUBROUTINE ***
                         *
                         ************************************************
                         *    ROUTINE TO ENABLE PARAMETER CHANGES
                         *    TO BE MADE. CALLED FROM POLL OR HALT.
                         ************************************************
                         *
                         *
0000 7C 0002  D QFIX     INC    F2        SET FLAG
                         *
0003 48                  ASLA             (A*2)
0004 B7 0001  D          STAA   CONST+1   OFFSET POINTER
0007 7F 0000  D          CLR    CONST
000A FE 0000  D          LDX    CONST     OFFSET INTO X
000D EE 00    A          LDX    J0,X      ADDR OF QUEST INTO X
                         *                   FROM JUMP TABLE
000F 6E 00    A          JMP    0,X       JUMP TO QUEST.
                         *
                         *
0011 7F 0002  D RTNPT    CLR    F2        CLR FLAG
                         *
0014 39                  RTS
                         *
                         *
                              XREF   J0
                         *
                              XDEF   QFIX,F2,RTNPT
                         *
                         *
0000                     DSCT
                         *
0000     0002  A CONST   RMB    2
0002     0001  A F2      RMB    1
                         *
                              END
ERRORS 00000
```

```
T
 NAM CT   VER.2   4-7-78   CLAVELL
 OPT REL
*
*  FILE NAMES:  &CT (S) / CT (R)
*
 TTL    ***  TOGGLE SUBROUTINE  ***
*
*    TOGGLES SHIFT LINE OF ELAPSED TIMER.
*    USED BY PROGM.
*
CT LDAA #$3E   SET BIT
 STAA P6BC
*
 LDX #23   DELAY
DEL DEX
 BNE DEL
*
 LDAA #$36     CLR BIT
 STAA P6BC
*
 RTS
*
 XREF P6BC
*
 XDEF CT
*
 END
```

SECTION IV:
Computing Routine

```
001   COMP    *** DATA COMPUTING ROUTINE ***

                          NAM     COMP      VER.3.2    ALLEN   2-7-79
                *            VER. 19    MODIFIED BY CLAVELL  4-27-79
                          OPT     REL
                *
                *  FILE NAMES:  &COMP(S) / COMP(R)
                *
                          TTL     *** DATA COMPUTING ROUTINE ***
                *
                *
00FC                      ORG     $00FC
                *
00FC    0080    A SABF    RMB     128       PLAIN SAMPLE PEAK AREA
                *
0000                      BSCT
                *
0000    0002    A MUL1    RMB     2
0002    0002    A MUL2    RMB     2
                *                    MULTIPLY & DIVIDE BUFF
0004    0002    A MUL3    RMB     2
0006    0002    A MUL4    RMB     2
        0002    B DIV2    EQU     MUL2
        0000    B DIV1    EQU     MUL1
        0004    B DIV3    EQU     MUL3
0008    0010    A CONBF   RMB     16        CONCENTRATION BUFF
0018    0002    A BINBUF  RMB     2         BIN TO ASCII CONV.
001A    0008    A LOWECO  RMB     8         LOW VALUE FOR EACH ELEMENT
0022    0008    A HIECO   RMB     8         HIGH VALUE FOR EACH ELEMEN
002A    0008    A SAAXA   RMB     8         BUFF FOR DIVISOR
                *
0000                      DSCT
                *
0000    0028    A PRBUF   RMB     40
                *
        0000    A YVAL    EQU     0         //Y VALUE OFFSET IN DATA A
        0002    A XVAL    EQU     2         //X VALUE OFFSET IN DATA A
0A36            *
0028    0004    A AREA    RMB     4
002C    0002    A BINTP1  RMB     2
002E    0002    A BINTP2  RMB     2
0030    0002    A TPC16   RMB     2
0032    0001    A ELNUM1  RMB     1
0033    0001    A ELNUM   RMB     1
0034    0001    A CYCLE   RMB     1
0035    0001    A SUBSMP  RMB     1
0036    0001    A SAMPLE  RMB     1
0037    0002    A CYNO    RMB     2         CURRENT Y ADDR
0039    0002    A CPTNO   RMB     2         CURRENT X ADDR
003B    0002    A FPTNO   RMB     2         X ADDR OF FIRST POINT ON P
003D    0002    A LPTNO   RMB     2         X ADDR OF LAST POINT ON PE
003F    0002    A CXVAL   RMB     2         CURRENT X VALUE
0041    0002    A FXVAL   RMB     2         FIRST PT IN PEAK
0043    0002    A LXVAL   RMB     2         LAST PT IN PEAK
0045    0001    A CNTR4   RMB     1
0046    0001    A HITCNT  RMB     1
0047    0001    A EQ      RMB     1
0048    0002    A PTREG1  RMB     2
004A    0002    A PTREG2  RMB     2
004C    0001    A FLAG1   RMB     1
```


Preceding Page Blank

```
002   COMP   *** DATA COMPUTING ROUTINE ***

004D    0001  A ONES    RMB    1
004E    0001  A TENS    RMB    1
004F    0001  A HUND    RMB    1
0050    0002  A TPT1    RMB    2
0052    0002  A TPT2    RMB    2
0054    0002  A TPT3    RMB    2
0056    0002  A TPT4    RMB    2
        0050  D X1      EQU    TPT1
        0052  D X2      EQU    TPT2
        0054  D Y1      EQU    TPT3
        0056  D Y2      EQU    TPT4
0058    0002  A EX      RMB    2
005A    0002  A Y       RMB    2
005C    0001  A SLOPE   RMB    1
005D    0001  A SABOFF  RMB    1
005E    0001  A BITHOU  RMB    1
005F    0001  A BIHUND  RMB    1
0060    0001  A BITENS  RMB    1
0061    0001  A BIONES  RMB    1
0062    0001  A RETF    RMB    1
0063    0001  A SIGN    RMB    1          /SIGN FLAG FOR MUL/DIV
0064    0004  A XA      RMB    4
0068    0002  A SAA     RMB    2
006A    0002  A SAC     RMB    2
006C    0002  A YA      RMB    2
006E    0002  A OR1     RMB    2
0070    0002  A R1      RMB    2
0072    0002  A R2      RMB    2
0074    0001  A ZESUP   RMB    1
0075    0001  A ERFLAG  RMB    1
0076    0001  A Z1F     RMB    1
0077    0001  A CADF    RMB    1
0078    0001  A COPF    RMB    1
0079    0001  A LEADF   RMB    1
007A    0018  A TIMBUF  RMB    24         HR , MIN & SEC BUFF
0092    0002  A TMBUF   RMB    2          POINTER TO TIMBUF
0094    0002  A SAVX    RMB    2
0096    0001  A SAVA    RMB    1
0097    0001  A INTCAL  RMB    1
        07C6  A YEND    EQU    $07C6     # OF DATA PT'S AT 1 PT/.6
                                        (0943 - 017D) = 07C6 (1990)
                *
                *
0000                    PSCT
                *
                **********************
                *ENTRY POINT AND DRIVER
                **********************
                *
0000 7D 0000    A COMPT  TST    CNTR2      VALID DATA FLAG
                *                            0 IF GOOD DATA
0003 27 10 0015          BEQ    CMPT0      ITS VALID
0005 7F 0076    D        CLR    Z1F        CLR ERROR FLAGS
0008 7F 0077    D        CLR    CADF
000B 7F 0078    D        CLR    COPF
000E 7F 0079    D        CLR    LEADF
                *
0011 BD 04CF    P        JSR    CLSABF
0014 39                  RTS                // NOT VALID
```

```
003   COMP    *** DATA COMPUTING ROUTINE ***

                      ********************
                      *FIND FIRST PEAK IN CURVE
                      ********************
                      *
0015 CE 0000  A CMPT0  LDX     #DATA     /FIRST POINT TO LOOK AT
0018 FF 0037  D        STX     CYNO
001B 7F 0063  D        CLR     SIGN
001E BD 0093  P CMPT1  JSR     FINDPK    /FIND FIRST PEAK IN CURVE
0021 7D 0033  D        TST     ELNUM     /THIS PEAK # 0=NONE
0024 27 0B 0031        BEQ     CMPT2     /NO MORE
0026 BD 0349  P        JSR     ADJUST    /ADJUST LINE AT BOTTOM
0029 BD 0400  P        JSR     INTGRT    /INTEGRATE ONE PEAK
002C BD 04DA  P        JSR     SAVRES    /SAVE INTEGRATED RESULTS
002F 20 ED 001E        BRA     CMPT1     /NEXT PEAK
0031 B6 0000  A CMPT2  LDAA    CNTR1
0034 B1 0000  A        CMPA    VPRECY    /NEED UP TO 8 CYCLES
0037 27 19 0052        BEQ     CMPT3     /READY TO CALC CONCS
                      *
0039 FE 0092  D        LDX     TMBUF     SAVE SAMPLE AQUIS. TIME
003C B6 0000  A        LDAA    HR
003F A7 00    A        STAA    0,X
0041 08                INX
0042 B6 0000  A        LDAA    MIN
0045 A7 00    A        STAA    0,X
0047 08                INX
0048 B6 0000  A        LDAA    SEC
004B A7 00    A        STAA    0,X
004D 08                INX
004E FF 0092  D        STX     TMBUF     UPDATE POINTER
                      *
0051 39                RTS               /NEED /MORE CYCLES
                      ********************
                      *CALCULATE CONCENTRATIONS
                      ********************
0052 7F 0035  D CMPT3  CLR     SUBSMP
0055 7F 0097  D        CLR     INTCAL
0058 7F 0074  D        CLR     ZESUP
005B CE 007A  D        LDX     #TIMBUF   INITIALIZE POINTER
005E FF 0092  D        STX     TMBUF
0061 7C 0097  D        INC     INTCAL
0064 BD 055B  P        JSR     CLSXA     CLR DIVISOR BUFF
0067 BD 0525  P CMPT4  JSR     CALCON    /CALCULATE CONCENTRATIONS
006A 7D 0097  D        TST     INTCAL
006D 27 08 0077        BEQ     CMPT6
006F 7F 0097  D        CLR     INTCAL
0072 7F 0035  D        CLR     SUBSMP
0075 20 F0 0067        BRA     CMPT4
                      *
0077 BD 070D  P CMPT6  JSR     PRTSMP    /PRINT SAMPLE
007A 7C 0036  D CMPT5  INC     SAMPLE
007D 7C 0035  D        INC     SUBSMP
0080 B6 0000  A        LDAA    VPRECY    //LAST SAMPLE
0083 4A                DECA
0084 B1 0035  D        CMPA    SUBSMP
0087 26 DE 0067        BNE     CMPT4     /NO, NEXT ONE
0089 BD 04CF  P        JSR     CLSABF
008C CE 007A  D        LDX     #TIMBUF
008F FF 0092  D        STX     TMBUF     REINITIALIZE POINTER
```

```
004   COMP   *** DATA COMPUTING ROUTINE ***

 0092 39                RTS              /YES, DONE WITH THIS GROUP
                        ******************
                        *FIND THE NEXT PEAK IN THE CURVE
                        *PUT
                        *FIRST POINT OFFSET IN FPTNO
                        *LAST POINT OFFSET IN LPTNO
                        *ELEMENT # IN ELNUM
                        *SET ELEMENT # TO 0 IF NO MORE PEAKS
                        ******************
                        *
                        *
                        *
 0093 BD 01CF  P FINDPK JSR     FNDPOS    //FIND A POSITIVE SLOPE
 0096 5D                TSTB
 0097 27 2B 00C4        BEQ     FIND3     //END OF CURVES
 0099 FE 0037  D        LDX     CYNO      /REMEMBER FIRST POINT
 009C 08                INX
 009D 08                INX
 009E FF 003B  D        STX     FPTNO     /IN THIS PEAK
 00A1 FE 003F  D        LDX     CXVAL
 00A4 FF 0041  D        STX     FXVAL
 00A7 BD 00F8  P        JSR     FNDEND    FIND END PT OF CURVE
 00AA 5D                TSTB
 00AB 27 17 00C4        BEQ     FIND3
 00AD FE 0037  D        LDX     CYNO      /LAST POINT IN THIS PEAK
 00B0 08                INX
 00B1 08                INX
 00B2 FF 003D  D        STX     LPTNO
 00B5 FE 003F  D        LDX     CXVAL     /LAST X AXIS VALUE
 00B8 FF 0043  D        STX     LXVAL
                        *
 00BB BD 02EF  P        JSR     WCHELM    DECIDE WHICH ELEMENT
                        *
 00BE 7D 0033  D        TST     ELNUM
 00C1 27 05 00C8        BEQ     FIND2
                        *
 00C3 39                RTS
 00C4 7F 0033  D FIND3  CLR     ELNUM     //NO MORE CURVES
 00C7 39                RTS
                        * CHECK FOR END OF DATA
 00C8 BD 00D1  P FIND2  JSR     ENDAT
 00CB 5D                TSTB
 00CC 27 F6 00C4        BEQ     FIND3     //OUT OF POINTS -- RETURN
 00CE 7E 0093  P        JMP     FINDPK    //KEEP LOOKING
                        *
                        * CHECK FOR END OF DATA
                        *  NEED AT LEAST 10 MORE VALID Y POINTS
                        *  NEED CPTNO <= YEND
                        * RETURN:
                        *   0 IF AT END OF DATA
                        *   1 IF NOT AT END OF DATA
                        *
                        *
                        *
 00D1 CE 0000  A ENDAT  LDX     #DATA
 00D4 FF 0000  A        STX     TDATA     CURRENT PT ADDR.
                        *                    MINUS STARTING PT ADDR
 00D7 B6 0037  D        LDAA    CYNO      = # OF PT'S CH'ED SO FAR
```

```
005   COMP   *** DATA COMPUTING ROUTINE ***

 00DA F6 0038  D          LDAB   CYNO+1
                *
 00DD F0 0001  A          SUBB   TDATA+1
 00E0 B2 0000  A          SBCA   TDATA
                *
 00E3 CB 28    A          ADDB   #40
 00E5 24 01 00E8          BCC    ENDAT1   //CARRY
 00E7 4C                  INCA
 00E8 CE 07C6  A ENDAT1   LDX    #YEND    //NUMBER OF ADDR.'S TO END
 00EB BD 0AA9  P          JSR    CMP16    //COMPARE THEM
 00EE C1 FF    A          CMPB   #$FF
 00F0 26 03 00F5          BNE    ENDAT2   //NOT END YET
 00F2 C6 00    A          LDAB   #0
 00F4 39                  RTS
 00F5 C6 01    A ENDAT2   LDAB   #1
 00F7 39                  RTS
                *
                *
                * MOVE ALONG THE PEAK FOUND BY FNDPOS
                * UNTIL FIND THE END OF PEAK THEN STORE
                * THAT POINT IN CYNO & CXVAL
                *
                *
 00F8 7F 004C  D FNDEND   CLR    FLAG1
 00FB 7F 0045  D C10      CLR    CNTR4
 00FE 7F 0046  D          CLR    HITCNT
 0101 7F 0047  D          CLR    EQ
                *
 0104 FE 0037  D C15      LDX    CYNO
 0107 FF 0048  D          STX    PTREG1
 010A 08                  INX             SET UP TEMP Y VALUE
 010B 08                  INX             BUFF
 010C 08                  INX
 010D 08                  INX
 010E FF 004A  D          STX    PTREG2
                *
 0111 7C 0045  D C20      INC    CNTR4    START LOOP
                *
 0114 BD 019E  P          JSR    NGPTCP
                *
 0117 81 02    A          CMPA   #2       PT1>PT2 ?
 0119 27 17 0132          BEQ    HIT3
 011B 7D 004C  D          TST    FLAG1    NO: DO WE HAVE 2 HITS YET?
 011E 26 04 0124          BNE    SKA1
 0120 81 01    A          CMPA   #1       NO: PT1=PT2 ?
 0122 26 3C 0160          BNE    C25
                *
 0124 B6 0047  D SKA1     LDAA   EQ       YES:
 0127 81 05    A          CMPA   #5       DO WE HAVE 5 PT'S IN LINE?
 0129 27 3E 0169          BEQ    SUC      YES:  SUCCESS,FOUND END
                *
 012B 7D 004C  D          TST    FLAG1    2 HITS YET?
 012E 26 14 0144          BNE    A10
 0130 20 0B 013D          BRA    C30
                *
 0132 7C 0046  D HIT3     INC    HITCNT
 0135 7F 0047  D          CLR    EQ
 0138 7D 004C  D          TST    FLAG1
```

```
006   COMP   *** DATA COMPUTING ROUTINE ***

 013B 26 07 0144          BNE     A10
                          *
 013D B6 0046  D C30      LDAA    HITCNT   2 HITS YET?
 0140 81 02    A          CMPA    #2       YES: THIS IS TOP OF PEAK
 0142 27 48 018C          BEQ     C40
                          *
 0144 BD 02AD  P A10      JSR     INCREG   MOVE TO NEXT PT
 0147 B6 0045  D          LDAA    CNTR4
 014A 81 0A    A          CMPA    #10      DONE 10 LOOPS YET?
 014C 26 C3 0111          BNE     C20      NO: LOOP AGAIN
                          *
 014E B6 0046  D          LDAA    HITCNT   YES: 5 OR MORE HITS?
 0151 81 04    A          CMPA    #4
 0153 2E 22 0177          BGT     C35
                          *
 0155 FE 0037  D          LDX     CYNO     NO: UPDATE X VALUE
 0158 EE 02    A          LDX     XVAL,X
 015A FF 003F  D          STX     CXVAL
                          *
 015D C6 01    A          LDAB    #1       SUCCESS - RETURN
 015F 39                  RTS
                          *
 0160 BD 02CC  P C25      JSR     NEXTPT   MOVE TO NEXT PT
 0163 5D                  TSTB             END OF DATA BUFF ?
 0164 27 25 018B          BEQ     C45      YES: DONE- RETURN
 0166 7E 00F8  P          JMP     FNDEND   START OVER
                          *
 0169 FE 004A  D SUC      LDX     PTREG2   FOUND ENDPT OF PEAK
 016C FF 0037  D          STX     CYNO
 016F EE 02    A          LDX     XVAL,X
 0171 FF 003F  D          STX     CXVAL
                          *
 0174 C6 01    A          LDAB    #1
 0176 39                  RTS
                          *
 0177 BD 00D1  P C35      JSR     ENDAT    END OF DATA BUFF?
                          *
 017A 5D                  TSTB
 017B 27 0E 018B          BEQ     C45
 017D FE 004A  D          LDX     PTREG2   NO: RESET CYNO TO
 0180 FF 0037  D          STX     CYNO     LAST PT
 0183 EE 02    A          LDX     XVAL,X
 0185 FF 003F  D          STX     CXVAL    UPDATE X VALUE
                          *
 0188 7E 00FB  P          JMP     C10      START OVER
                          *
 018B 39          C45     RTS
                          *
 018C FE 004A  D C40      LDX     PTREG2   RESET CYNO
 018F FF 0037  D          STX     CYNO
 0192 7C 004C  D          INC     FLAG1
 0195 7F 0045  D          CLR     CNTR4
 0198 7F 0047  D          CLR     EQ
 019B 7E 0104  P          JMP     C15
                          *
                          *
 019E FE 0048  D NGPTCP   LDX     PTREG1
 01A1 A6 00    A          LDAA    0,X
```

```
007   COMP   *** DATA COMPUTING ROUTINE ***

01A3 E6 01    A         LDAB   1,X
01A5 FE 004A  D         LDX    PTREG2
01A8 EE 00    A         LDX    0,X
                *
01AA BD 0AA9  P         JSR    CMP16
                *
01AD 5D                 TSTB
01AE 27 09 01B9         BEQ    C55       X=AB ?
01B0 C1 01    A         CMPB   #1        X>AB
01B2 27 03 01B7         BEQ    C50
01B4 86 02    A         LDAA   #2        X<AB
                *
01B6 39                 RTS
                *
01B7 4F           C50   CLRA
01B8 39                 RTS
                *
01B9 7C 0047  D C55     INC    EQ
01BC 86 01    A         LDAA   #1
01BE 39                 RTS
                *
                *
01BF FE 0039  D NEXPNT  LDX    CPTNO
01C2 08                 INX
01C3 08                 INX
01C4 08                 INX
01C5 08                 INX
01C6 FF 0039  D         STX    CPTNO
                *
01C9 EE 00    A         LDX    0,X
01CB FF 003F  D         STX    CXVAL
                *
01CE 39                 RTS
                *
                *
                *
                *
                * SEARCH FOR THE BEGINNING OF A PEAK DEFINED
                *  BY 5  POINTS WITH POSITIVE SLOPE OUT OF 10
                *  CONSECUTIVE POINTS STARTING AT CURRENT POINT.
                *
01CF 7F 0045  D FNDPOS  CLR    CNTR4
01D2 7F 0046  D         CLR    HITCNT
01D5 7F 0047  D         CLR    EQ
                *
01D8 FE 0037  D         LDX    CYNO
01DB FF 0048  D         STX    PTREG1    INITIALIZE PTREG
01DE 08                 INX
01DF 08                 INX               & PTREG2 TO 1ST
01E0 08                 INX               & 2ND PT'S TO
01E1 08                 INX               COMPARE
01E2 FF 004A  D         STX    PTREG2
                *
01E5 7C 0045  D ENT     INC    CNTR4
                *
01E8 BD 0234  P         JSR    PTCMP     DO PT COMPARISONS
                *
01EB 81 02    A         CMPA   #2        A=2?
```

```
008   COMP   *** DATA COMPUTING ROUTINE ***

 01ED 27 0D 01FC          BEQ    HIT        YES: INC HIT COUNTER
 01EF 81 01     A         CMPA   #1         A=1?
 01F1 26 17 020A          BNE    FP         NO: CHK FOR BUFF END
                  *
 01F3 B6 0047   D         LDAA   EQ         EQ = 3?
 01F6 81 03     A         CMPA   #3
 01F8 27 0D 0207          BEQ    UPD        YES: UPDATE FIRST PT & STA
 01FA 20 06 0202          BRA    ICREG
                  *
 01FC 7C 0046   D HIT     INC    HITCNT
 01FF 7F 0047   D         CLR    EQ
                  *
 0202 BD 02AD   P ICREG   JSR    INCREG
 0205 20 0D 0214          BRA    HIT1
                  *
 0207 BD 02A1   P UPD     JSR    LNUPDT     MOVE TO NEXT PT
                  *
 020A BD 00D1   P FP      JSR    ENDAT      CHK FOR END OF DATA BUFF
 020D 5D                  TSTB
 020E 26 01 0211          BNE    F1         NOT DONE YET
 0210 39                  RTS               EMPTY-RETURN
 0211 7E 01CF   P F1      JMP    FNDPOS
                  *
                  *                         ONE PT OVER
 0214 B6 0046   D HIT1    LDAA   HITCNT     DO WE HAVE 5 PT'S?
 0217 81 05     A         CMPA   #5         WITH POS SLOPE?
 0219 26 0B 0226          BNE    LOOP1      NO: MAKE ANOTHER LOOP
                  *
 021B C6 FF     A         LDAB   #$FF       YES: SUCCESS
 021D FE 0037   D         LDX    CYNO
 0220 EE 02     A         LDX    XVAL,X     UPDATE X VALUE
 0222 FF 003F   D         STX    CXVAL
                  *
 0225 39                  RTS
                  *
 0226 B6 0045   D LOOP1   LDAA   CNTR4      HAVE WE CHK'ED 10 PT'S?
 0229 81 0A     A         CMPA   #10
 022B 27 03 0230          BEQ    NP1        YES START AGAIN:
                  *
 022D 7E 01E5   P         JMP    ENT
                  *
 0230 BD 02C2   P NP1     JSR    FNDP2      YES: MOVE OVER 1 PT
                  *                        AND REPEAT
 0233 39                  RTS
                  *
                  *
 0234 FE 004A   D PTCMP   LDX    PTREG2
 0237 A6 00     A         LDAA   0,X
 0239 E6 01     A         LDAB   1,X
 023B FE 0048   D         LDX    PTREG1
 023E EE 00     A         LDX    0,X
                  *
 0240 BD 0AA9   P         JSR    CMP16
                  *
 0243 5D                  TSTB
 0244 27 0B 0251          BEQ    EQINC      X=AB
 0246 C1 01     A         CMPB   #1         X>AB
 0248 27 03 024D          BEQ    CK10
```

```
009   COMP   *** DATA COMPUTING ROUTINE ***

024A 86 02    A         LDAA   #2        X<AB
024C 39                 RTS
                *
024D BD 0257  P CK10    JSR    CKNXPT
0250 39                 RTS
                *
0251 7C 0047  D EQINC   INC    EQ
0254 86 01    A         LDAA   #1
0256 39                 RTS
                *
                *
0257 FE 004A  D CKNXPT  LDX    PTREG2
025A 08                 INX               GET NEXT Y VALUE ADR
025B 08                 INX
025C 08                 INX
025D 08                 INX
025E FF 004A  D         STX    PTREG2
                *
0261 FE 004A  D         LDX    PTREG2     COMPARE 1ST & 3RD PT'S
0264 A6 00    A         LDAA   0,X
0266 E6 01    A         LDAB   1,X
0268 FE 0048  D         LDX    PTREG1
026B EE 00    A         LDX    0,X
                *
026D BD 0AA9  P         JSR    CMP16
                *
0270 5D                 TSTB
0271 27 1E 0291         BEQ    EQINC1     X=AB -  RETURN 1 IN A
0273 C1 FF    A         CMPB   #$FF       X<AB - RETURN 2 IN A
0275 27 0D 0284         BEQ    HIT2
                *                          X>AB -  RETURN 0 IN A
0277 FE 004A  D         LDX    PTREG2     RESET CYNO
027A FF 0037  D         STX    CYNO
027D EE 02    A         LDX    XVAL,X
027F FF 003F  D         STX    CXVAL
0282 4F                 CLRA
0283 39                 RTS
                *
0284 FE 0048  D HIT2    LDX    PTREG1
0287 08                 INX
0288 08                 INX
0289 08                 INX
028A 08                 INX
028B FF 0048  D         STX    PTREG1
                *
028E 86 02    A         LDAA   #2
0290 39                 RTS
                *
0291 7C 0047  D EQINC1  INC    EQ
0294 FE 0048  D         LDX    PTREG1
0297 08                 INX
0298 08                 INX
0299 08                 INX
029A 08                 INX
029B FF 0048  D         STX    PTREG1
029E 86 01    A         LDAA   #1
02A0 39                 RTS
                *
```

```
010   COMP   *** DATA COMPUTING ROUTINE ***

                     *
02A1 FE 0048  D LNUPDT LDX    PTREG1
02A4 FF 0037  D        STX    CYNO      RESET CYNO
                     *
02A7 EE 02    A        LDX    XVAL,X
02A9 FF 003F  D        STX    CXVAL     UPDATE X VALUE
                     *
02AC 39                RTS
                     *
                     *
02AD FE 0048  D INCREG LDX    PTREG1
02B0 08                INX
02B1 08                INX
02B2 08                INX
02B3 08                INX
02B4 FF 0048  D        STX    PTREG1
                     *
02B7 FE 004A  D        LDX    PTREG2
02BA 08                INX
02BB 08                INX
02BC 08                INX
02BD 08                INX
02BE FF 004A  D        STX    PTREG2
                     *
02C1 39                RTS
                     *
                     *
02C2 BD 02CC  P FNDP2  JSR    NEXTPT
02C5 5D                TSTB
02C6 26 01 02C9        BNE    ME2
02C8 39                RTS
02C9 7E 01CF  P ME2    JMP    FNDPOS
                     *
                     *GO TO NEXT POINT (TO RIGHT ALONG X AXIS)
02CC FE 0037  D NEXTPT LDX    CYNO      /CURRENT POINT #
02CF 08                INX
02D0 08                INX
02D1 08                INX
02D2 08                INX
02D3 FF 0037  D        STX    CYNO      /NEXT POINT
02D6 EE 02    A        LDX    XVAL,X
02D8 FF 003F  D        STX    CXVAL
02DB BD 00D1  P        JSR    ENDAT     //CHECK FOR END OF YO1
02DE 39                RTS
                     ********************
                     *GO TO PREVIOUS POINT (TO LEFT ALONG X AXIS)
                     ********************
02DF FE 0039  D LASTPT LDX    CPTNO
02E2 09                DEX
02E3 09                DEX
02E4 09                DEX
02E5 09                DEX
02E6 FF 0039  D        STX    CPTNO
02E9 EE 00    A        LDX    0,X
02EB FF 003F  D        STX    CXVAL
02EE 39                RTS
                     ********************
                     *DECIDE WHICH ELEMENT THE CURRENT
```

```
011   COMP    *** DATA COMPUTING ROUTINE ***

                *PEAK REPRESENTS IF ANY
                *RETURN:
                * 0 - NOT A VALID PEAK
                * 2 - ZINC 0
                * 4 - CADMIUM 2
                * 6 - LEAD 4
                * 8 - COPPER 6
                **********
02EF 86 08    A WCHELM LDAA   #8
02F1 B7 0033  D        STAA   ELNUM     /TRY ALL 4 ELEMENTS
02F4 CE 0006  A        LDX    #6
02F7 FF 0056  D WCHE0  STX    TPT4      /SAVE ELEMENT INDEX
02FA EE 1A    B        LDX    LOWECO,X  /LOW POSSIBLE
02FC B6 0041  D        LDAA   FXVAL     /FISRT POINT IN PEAK
02FF F6 0042  D        LDAB   FXVAL+1
0302 BD 0AA9  P        JSR    CMP16
0305 C1 01    A        CMPB   #1        IF =1 NO FIT
0307 27 19 0322        BEQ    WCHE1     MUST BE (X<=AB)
0309 FE 0056  D        LDX    TPT4
030C EE 22    B        LDX    HIECO,X   /HIGH POSSIBLE
030E B6 0043  D        LDAA   LXVAL
0311 F6 0044  D        LDAB   LXVAL+1
                *
0314 4D                TSTA
0315 26 03 031A        BNE    WCHE6     STILL HAVE DATA
0317 5D                TSTB
0318 27 2B 0345        BEQ    WCHE5     NO MORE DATA: STOP
                *
031A BD 0AA9  P WCHE6  JSR    CMP16     MUST BE (X> OR = AB)
031D C1 FF    A        CMPB   #$FF
031F 27 01 0322        BEQ    WCHE1
0321 39        WCHE2   RTS
0322 7D 0000  A WCHE1  TST    ZINK
0325 27 0A 0331        BEQ    WCHE3
                *
0327 7A 0033  D        DEC    ELNUM
032A 7A 0033  D        DEC    ELNUM
032D 27 F2 0321        BEQ    WCHE2     /NOT A VALID PEAK
032F 20 0D 033E        BRA    WCHE4
                *
0331 7A 0033  D WCHE3  DEC    ELNUM     IF NOT DOING ZN
0334 7A 0033  D        DEC    ELNUM     DON'T CMP. RANGE
0337 B6 0033  D        LDAA   ELNUM
033A 81 02    A        CMPA   #2
033C 27 07 0345        BEQ    WCHE5
                *
033E FE 0056  D WCHE4  LDX    TPT4
0341 09                DEX
0342 09                DEX
0343 20 B2 02F7        BRA    WCHE0
0345 7F 0033  D WCHE5  CLR    ELNUM
0348 39                RTS
                *
                **********
                *ADJUST THE LINE AT THE BOTTOM OF CURVE
                *TO INSURE THAT NO POINTS ON THE
                *CURVE ARE BELOW THE LINE
                *ASSUMES:
```

```
012   COMP   *** DATA COMPUTING ROUTINE ***

                     * FPTNO CONTAINS THE FIRST POINT ADDR
                     * FXVAL CONTAINS THE FIRST X VALUE
                     * LPTNO CONTAINS THE LAST POINT ADDR
                     * LXVAL CONTAINS THE LAST X VALUE
                     *RETURNS:
                     * FPTNO,FXVAL,LPTNO,LXVAL DEFINING
                     *THE END POINTS OF THE LINE AT THE BOTTOM
                     *OF THE CURVE WITH NO POINTS MISSING.
                     ************************************
 0349 FE 003B  D ADJUST LDX    FPTNO    /LEFT END PT OF LINE
 034C FF 0039  D        STX    CPTNO
 034F FE 0041  D        LDX    FXVAL    /X VALUE AT LEFT END OF LI
 0352 FF 003F  D        STX    CXVAL    /
 0355 BD 01BF  P        JSR    NEXPNT   /LOOK AT NEXT PT TO RIGHT
 0358 BD 0390  P        JSR    ABLINE   /IS THIS PT ABOVE THE LINE
 035B 4D                TSTA            /#1=TRUE
 035C 26 0E 036C        BNE    ADJ5     /ADJUST RIGHT END
 035E FE 0039  D        LDX    CPTNO    /NEW LEFT END
 0361 FF 003B  D        STX    FPTNO
 0364 FE 003F  D        LDX    CXVAL
 0367 FF 0041  D        STX    FXVAL
 036A 20 DD 0349        BRA    ADJUST   /TRY AGAIN
 036C FE 003D  D ADJ5   LDX    LPTNO    /RIGHT END PT
 036F FF 0039  D        STX    CPTNO
 0372 FE 0043  D        LDX    LXVAL
 0375 FF 003F  D        STX    CXVAL
 0378 BD 02DF  P        JSR    LASTPT   /MOVE ONE PT TO LEFT
 037B BD 0390  P        JSR    ABLINE   /IS THIS PT ABOVE THE LINE
 037E 4D                TSTA
 037F 26 0E 038F        BNE    ADJ6
 0381 FE 0039  D        LDX    CPTNO
 0384 FF 003D  D        STX    LPTNO
 0387 FE 003F  D        LDX    CXVAL
 038A FF 0043  D        STX    LXVAL
 038D 20 DD 036C        BRA    ADJ5
 038F 39          ADJ6  RTS
                     ********************
                     *DETERMINES IF THE POINT DEFINED BY
                     *CPTNO AND CXVAL IS ABOVE THE LINE
                     *DETERMINED BY FPTNO,FXVAL AND LPTNO,LXVAL
                     *RETURN 1=TRUE 0=FALSE IN ACCA
                     *Y VALUE ON LINE IS LEFT IN Y
                     ********************
 0390 FE 003B  D ABLINE LDX    FPTNO    /GET Y1
 0393 09                DEX             1ST Y VALUE
 0394 09                DEX
 0395 EE 00    A        LDX    0,X
 0397 FF 0054  D        STX    Y1
 039A FE 003D  D        LDX    LPTNO    /GET Y2
 039D 09                DEX             LAST Y VALUE
 039E 09                DEX
 039F EE 00    A        LDX    0,X
 03A1 FF 0056  D        STX    Y2
                     *
 03A4 B6 0057  D        LDAA   Y2+1     /FORM (Y2-Y1)
 03A7 B0 0055  D        SUBA   Y1+1
 03AA 97 01    B        STAA   MUL1+1
 03AC B6 0056  D        LDAA   Y2
```

```
013   COMP    *** DATA COMPUTING ROUTINE ***

 03AF B2 0054  D          SBCA    Y1
 03B2 97 00    B          STAA    MUL1
 03B4 B6 0040  D          LDAA    CXVAL+1   /FORM (XC-X1)
 03B7 B0 0042  D          SUBA    FXVAL+1
 03BA 97 05    B          STAA    MUL3+1
 03BC B6 003F  D          LDAA    CXVAL
 03BF B2 0041  D          SBCA    FXVAL
 03C2 97 04    B          STAA    MUL3
                  *
 03C4 BD 0997  P          JSR     MUL       /MULTIPLY (Y2-Y1)(XC-X1)
                  *
 03C7 B6 0044  D          LDAA    LXVAL+1   /FORMS (X2-X1)
 03CA B0 0042  D          SUBA    FXVAL+1
 03CD F6 0043  D          LDAB    LXVAL     /DIVISOR IN A(LO) &
 03D0 F2 0041  D          SBCB    FXVAL     / B (HI)
                  *
 03D3 BD 09F1  P          JSR     DDIV      /DIVIDE(Y2-Y1)(X-X1) BY (X
                  *
 03D6 96 05    B          LDAA    DIV3+1    /ADD Y1 TO RESULT
 03D8 BB 0055  D          ADDA    Y1+1
 03DB B7 005B  D          STAA    Y+1       (SLOPE * X)
 03DE 96 04    B          LDAA    DIV3
 03E0 B9 0054  D          ADCA    Y1
 03E3 B7 005A  D          STAA    Y         STORE IN 'Y'
 03E6 FE 0039  D          LDX     CPTNO     /CURRENT PT #
 03E9 09                  DEX               GET Y VALUE AT THIS PT
 03EA 09                  DEX
 03EB EE 00    A          LDX     0,X
 03ED B6 005A  D          LDAA    Y
 03F0 F6 005B  D          LDAB    Y+1
 03F3 BD 0AA9  P          JSR     CMP16     /COMPARE TO CALCULATED VAL
 03F6 C1 FF    A          CMPB    #$FF
 03F8 26 03 03FD          BNE     ABL1      /POINT IS ABOVE OR EQUAL T
 03FA 86 00    A          LDAA    #0        /RETURN FALSE IF PT IS < L
 03FC 39                  RTS
 03FD 86 01    A ABL1     LDAA    #1        /RETURN TRUE IF PT > OR =
 03FF 39                  RTS
                  ********************
                  *CALCULATE THE AREA UNDER THE FUNCTION CURVE
                  *DELINEATED BY FPTNO,FXVAL AND LPTNO,LXVAL.
                  *THIS ROUTINE USES THE TRAPEZOIDAL RULE FOR
                  *STEP-WISE INTEGRATION APPROXIMATION.
                  ********************
 0400 FE 003B  D INTGRT LDX     FPTNO     /START OF POSITION IN CURV
 0403 FF 0039  D          STX     CPTNO
 0406 FE 0041  D          LDX     FXVAL
 0409 FF 003F  D          STX     CXVAL
 040C 7F 005C  D          CLR     SLOPE
 040F FE 003D  D          LDX     LPTNO     DETERMINE BASE LINE
 0412 09                  DEX               SLOPE SIGN
 0413 09                  DEX
 0414 A6 00    A          LDAA    0,X
 0416 E6 01    A          LDAB    1,X       LAST Y VALUE
                  *
 0418 FE 003B  D          LDX     FPTNO
 041B 09                  DEX
 041C 09                  DEX
 041D EE 00    A          LDX     0,X       1ST Y VALUE
```

```
014   COMP   *** DATA COMPUTING ROUTINE ***

                     *
 041F BD 0AA9  P          JSR     CMP16
 0422 C1 01    A          CMPB    #1
 0424 26 03 0429          BNE     INTG1     /SLOPE IS POSITIVE
 0426 7C 005C  D          INC     SLOPE     /SLOPE IS NEGATIVE
 0429 CE 0000  A INTG1    LDX     #0
 042C FF 0028  D          STX     AREA      /CLEAR AREA BUFFER
 042F FF 002A  D          STX     AREA+2
 0432 BD 0446  P INTG2    JSR     GTAREA    /CALC AREA OF ONE TRAPEZOI
 0435 FE 0039  D          LDX     CPTNO
 0438 B6 003D  D          LDAA    LPTNO
 043B F6 003E  D          LDAB    LPTNO+1
 043E BD 0AA9  P          JSR     CMP16     /ARE WE THRU?
 0441 C1 FF    A          CMPB    #$FF
 0443 27 ED 0432          BEQ     INTG2     /NO, DO NEXT TRAPEZOID
 0445 39                  RTS               /THRU WITH THIS CURVE
                     ********************
                     *THIS ROUTINE CALCULATES THE AREA OF ONE
                     *TRAPEZOID DEFINED BY CPTNO AND CPTNO+1 AND
                     *ADDS THIS CALCULATED AREA TO AREA
                     **THE TRAPEZOIDAL RULE FOR STEP-WISE INTEGRATION
                     ********************
 0446 7D 005C  D GTAREA TST     SLOPE     /IS SLOPE OF LINE POSITIVE
 0449 27 03 044E          BEQ     GTAR1     /YES
 044B BD 01BF  P          JSR     NEXPNT    /NEGATIVE - USE RIGHT SIDE
 044E BD 0390  P GTAR1    JSR     ABLINE    /CALC Y VALUE ON BASE LINE
 0451 7D 005C  D          TST     SLOPE
 0454 27 03 0459          BEQ     GTAR2
 0456 BD 02DF  P          JSR     LASTPT    /BACK TO LEFT SIDE
 0459 FE 0039  D GTAR2    LDX     CPTNO     / CURRENT X ADDR
 045C 09                  DEX               Y-ADDR.
 045D 09                  DEX
 045E A6 01    A          LDAA    1,X       /LOW  BYTE (Y VALUE)
 0460 B0 005B  D          SUBA    Y+1       /SUBTRACT BASE LINE
 0463 B7 0055  D          STAA    Y1+1      /Y VALUE
 0466 A6 00    A          LDAA    0,X       /HIGH  BYTE Y VALUE
 0468 B2 005A  D          SBCA    Y         /BASE LINE UPPER BYTE
 046B B7 0054  D          STAA    Y1        /THIS IS F(X0)
 046E BD 01BF  P          JSR     NEXPNT    /RIGHT SIDE
 0471 FE 0039  D          LDX     CPTNO
 0474 09                  DEX               Y ADDR
 0475 09                  DEX
 0476 A6 01    A          LDAA    1,X       /LOWER BYTE
 0478 B0 005B  D          SUBA    Y+1       /SUBTRACT  BASE LINE
 047B B7 0057  D          STAA    Y2+1
 047E A6 00    A          LDAA    0,X       /UPPER
 0480 B2 005A  D          SBCA    Y         /BASE LINE UPPER
 0483 B7 0056  D          STAA    Y2        /THIS IS F(X1)
 0486 B6 0055  D          LDAA    Y1+1      /F(X0) LOWER
 0489 BB 0057  D          ADDA    Y2+1      /PLUS F(X1) LOWER
 048C 97 01    B          STAA    MUL1+1
 048E B6 0054  D          LDAA    Y1        /F(X0) UPPER
 0491 B9 0056  D          ADCA    Y2        /PLUS F(X1) UPPER
 0494 97 00    B          STAA    MUL1      /F(X0) + F(X1)
 0496 FE 0039  D          LDX     CPTNO     //CURRENT POINT POINTER
 0499 A6 05    A          LDAA    5,X       //CALC H*2 = X2-X1
 049B A0 01    A          SUBA    1,X
 049D 97 05    B          STAA    MUL3+1
```

```
015   COMP   *** DATA COMPUTING ROUTINE ***

 049F A6 04     A          LDAA   4,X
 04A1 A2 00     A          SBCA   0,X
 04A3 97 04     B          STAA   MUL3
 04A5 74 0004   B          LSR    MUL3       //H = (X2-X1)/2
 04A8 76 0005   B          ROR    MUL3+1
 04AB BD 0997   P          JSR    MUL        //AREA = H/2*(F(X1)+F(X2))
                         *NOW ADD THIS TRAPEZOID AREA INTO AREA
 04AE 96 05     B          LDAA   MUL3+1
 04B0 BB 002B   D          ADDA   AREA+3
 04B3 B7 002B   D          STAA   AREA+3
 04B6 96 04     B          LDAA   MUL3
 04B8 B9 002A   D          ADCA   AREA+2
 04BB B7 002A   D          STAA   AREA+2
 04BE 96 03     B          LDAA   MUL2+1
 04C0 B9 0029   D          ADCA   AREA+1
 04C3 B7 0029   D          STAA   AREA+1
 04C6 96 02     B          LDAA   MUL2
 04C8 B9 0028   D          ADCA   AREA
 04CB B7 0028   D          STAA   AREA
 04CE 39                 GTAR3  RTS
                         *
                         * CLEAR THE PLAIN SAMPLE PEAK AREA BUFFERS
                         *
 04CF 86 00     A       CLSABF LDAA   #0
 04D1 CE 0080   A          LDX    #128
 04D4 A7 FB     A       CLSA1  STAA   SABF-1,X
 04D6 09                   DEX
 04D7 26 FB 04D4           BNE    CLSA1
 04D9 39                   RTS
                         ********************
                         *SAVE THE INTEGRATION RESULTS IN PROPER BUFFER
                         ********************
 04DA B6 0033   D       SAVRES LDAA   ELNUM      RSSET ELNUM FOR PROPER
 04DD 81 02     A          CMPA   #2          OFFSET
 04DF 26 05 04E6           BNE    SAV1
                         *
 04E1 7F 0033   D          CLR    ELNUM       SET ZN = 0
 04E4 20 16 04FC           BRA    SAV4
                         *
 04E6 81 04     A       SAV1   CMPA   #4
 04E8 26 02 04EC           BNE    SAV2        CD OK
 04EA 20 10 04FC           BRA    SAV4
                         *
 04EC 81 06     A       SAV2   CMPA   #6
 04EE 26 07 04F7           BNE    SAV3
                         *
 04F0 8B 02     A          ADDA   #2          PB = 8
 04F2 B7 0033   D          STAA   ELNUM
 04F5 20 05 04FC           BRA    SAV4
                         *
 04F7 8B 04     A       SAV3   ADDA   #4
 04F9 B7 0033   D          STAA   ELNUM       CU = 12
                         *
 04FC 7F 0032   D       SAV4   CLR    ELNUM1      /HIGH OF INDEX
 04FF F6 0000   A          LDAB   CNTR1       //CYCLE #
 0502 5A                   DECB               NO 0 CYCLE
 0503 58                   ASLB               //*2
 0504 58                   ASLB               //*4
```

```
016   COMP   *** DATA COMPUTING ROUTINE ***

 0505 58                ASLB              //*8
 0506 58                ASLB              //*16
 0507 FB 0033  D        ADDB    ELNUM
 050A F7 0033  D        STAB    ELNUM
 050D FE 0032  D        LDX     ELNUM1
 0510 B6 0028  D        LDAA    AREA
 0513 A7 FC    A        STAA    SABF,X
 0515 B6 0029  D        LDAA    AREA+1
 0518 A7 FD    A        STAA    SABF+1,X
 051A B6 002A  D        LDAA    AREA+2
 051D A7 FE    A        STAA    SABF+2,X AREA IS IN 32 BITS
 051F B6 002B  D        LDAA    AREA+3
 0522 A7 FF    A        STAA    SABF+3,X
 0524 39                RTS
                      *CALCULATE THE CONCENTRATIONS FOR THE THREE
                      *PLAIN SAMPLE PEAK AREAS THAT ARE NON-ZERO
                      *IN SABF. PUT RESULTS IN CONBF
 0525 BD 0551  P CALCON JSR     CLCNBF   / CLEAR CONCENTRATION BUFF
 0528 7D 0097  D        TST     INTCAL
 052B 27 08 0535        BEQ     CALC0
                      *
 052D B6 0000  A        LDAA    VPRECY
 0530 4A                DECA
 0531 4A                DECA
 0532 B7 0035  D        STAA    SUBSMP   OFFSET
                      *
 0535 86 00    A CALC0  LDAA    #0
 0537 B7 0033  D CALCO  STAA    ELNUM
 053A F6 0035  D        LDAB    SUBSMP
 053D 58                ASLB    *2
 053E 58                ASLB    *4
 053F 58                ASLB    *8
 0540 58                ASLB    *16
 0541 F7 005D  D CALC1  STAB    SABOFF
 0544 BD 0565  P        JSR     CL1CON   /TRY TO CALCULATE THIS CON
 0547 B6 0033  D        LDAA    ELNUM
 054A 8B 04    A        ADDA    #4
 054C 81 0C    A        CMPA    #12
 054E 2F E7 0537        BLE     CALCO    /NEXT
 0550 39                RTS
                      *CLEAR THE CONCENTRATION BUFFER
 0551 4F              CLCNBF CLRA
 0552 CE 0010  A        LDX     #16
 0555 A7 07    B CLCNB1 STAA    CONBF-1,X
 0557 09                DEX
 0558 26 FB 0555        BNE     CLCNB1
 055A 39                RTS
                      *
                      *CLEAR THE DIVISOR BUFFER (SAAXA)
                      *
 055B 4F              CLSXA  CLRA
 055C CE 0008  A        LDX     #8
 055F A7 29    B CLSXA1 STAA    SAAXA-1,X
 0561 09                DEX
 0562 26 FB 055F        BNE     CLSXA1
 0564 39                RTS
                      *
                      *
```

```
017   COMP    *** DATA COMPUTING ROUTINE ***

                  *CALCULATE CONCENTRATION FOR ONE ELEMENT IF
                  *AREA IS NOT ZERO
                  *
                  *
0565 B6 0033  D CL1CON LDAA   ELNUM     CHK IF STD CON IS 0
                  *                       IF IT IS- RETURN
0568 16                 TAB             A TO B
0569 54                 LSRB            B/2
056A F7 0059  D         STAB   EX+1
056D 7F 0058  D         CLR    EX
0570 FE 0058  D         LDX    EX       OFFSET
0573 E6 00    A         LDAB   STCN1H,X GET STD CON VALUE
0575 5D                 TSTB
0576 26 01 0579         BNE    CL1C2
0578 39                 RTS
                  *
0579 B6 005D  D CL1C2  LDAA   SABOFF    /BUFFER OFFSET
057C BB 0033  D         ADDA   ELNUM    /ELEMENT AREA OFFSET
057F B7 0059  D         STAA   EX+1
0582 7F 0058  D         CLR    EX
0585 FE 0058  D         LDX    EX
0588 6D FC    A         TST    SABF,X
058A 26 0D 0599         BNE    CL1C1
058C 6D FD    A         TST    SABF+1,X
058E 26 09 0599         BNE    CL1C1
0590 6D FE    A         TST    SABF+2,X
0592 26 05 0599         BNE    CL1C1
0594 6D FF    A         TST    SABF+3,X
0596 26 01 0599         BNE    CL1C1
0598 39                 RTS             /NO AREA
0599 A6 FC    A CL1C1  LDAA   SABF,X
059B B7 0064  D         STAA   XA
059E A6 FD    A         LDAA   SABF+1,X
05A0 B7 0065  D         STAA   XA+1
05A3 A6 FE    A         LDAA   SABF+2,X
05A5 B7 0066  D         STAA   XA+2
05A8 A6 FF    A         LDAA   SABF+3,X
05AA B7 0067  D         STAA   XA+3     //UNKNOWN PEAK AREA IN XA
                  *
05AD 7D 0097  D         TST    INTCAL
05B0 27 3B 05ED         BEQ    XADIV
                  *
05B2 B6 0000  A         LDAA   VPRECY   //STD ADD CYCLE #
05B5 4A                 DECA            ADJUST TO PROPER POSITION
05B6 48                 ASLA
05B7 48                 ASLA
05B8 48                 ASLA
05B9 48                 ASLA
05BA BB 0033  D         ADDA   ELNUM    //CURRENT ELEMENT
05BD B7 0059  D         STAA   EX+1
05C0 7F 0058  D         CLR    EX       OFFSET TO POINT TO
05C3 FE 0058  D         LDX    EX       STD PEAK AREA
                  *
05C6 A6 FF    A         LDAA   SABF+3,X USE LO 16 BITS
05C8 B0 0067  D         SUBA   XA+3
05CB B7 0096  D         STAA   SAVA     SAVE DIVISOR
                  *
05CE E6 FE    A         LDAB   SABF+2,X SAA-XA   LO BYTES
```

```
018   COMP    *** DATA COMPUTING ROUTINE ***

 05D0 F2 0066  D          SBCB    XA+2      DIVISOR
 05D3 2D 15 05EA          BLT     E1        IF SAA-XA IS NEG: ERROR
                  *
 05D5 B6 0033  D          LDAA    ELNUM     GET OFFSET
 05D8 44                  LSRA              /2
 05D9 B7 0059  D          STAA    EX+1
 05DC 7F 0058  D          CLR     EX
 05DF FE 0058  D          LDX     EX        OFFSET INTO X
 05E2 B6 0096  D          LDAA    SAVA      RESTORE A
                  *
 05E5 A7 2B    B          STAA    SAAXA+1,X
 05E7 E7 2A    B          STAB    SAAXA,X   SAVE DIVISOR'S
 05E9 39                  RTS
                  *
 05EA 7E 06EE  P E1       JMP     ER1       ERROR MESSAGE
                  *
 05ED B6 0033  D XADIV    LDAA    ELNUM
 05F0 44                  LSRA
 05F1 B7 0059  D          STAA    EX+1
 05F4 7F 0058  D          CLR     EX
 05F7 FE 0058  D          LDX     EX
                  *
 05FA A6 2B    B          LDAA    SAAXA+1,X
 05FC E6 2A    B          LDAB    SAAXA,X   DIVISOR FOR CURRENT PEAK
                  *
 05FE 5D                  TSTB              CHK IF DIVISOR 0
 05FF 26 04 0605          BNE     OK
 0601 4D                  TSTA
 0602 26 01 0605          BNE     OK
 0604 39                  RTS
                  *
 0605 FE 0064  D OK       LDX     XA
 0608 DF 02    B          STX     DIV2
 060A FE 0066  D          LDX     XA+2
 060D DF 04    B          STX     DIV3      //XA IS DIVIDEND
                  *
 060F BD 09F1  P          JSR     DDIV      // XA/(SAA-XA)
                  *
                  *
                  *     QUOT. = DIV3 & REM = DIV2
                  *      OPERATE ON RESULT SO AS NOT TO LOSE
                  *      REMAINDER FOR NEXT MULTIPLICATION
                  *
                  *
 0612 DE 02    B          LDX     DIV2      SAVE REM
 0614 FF 006E  D          STX     OR1
                  *
 0617 7D 0004  B          TST     DIV3      CHK IF QUOT < 99
 061A 26 08 0624          BNE     ER0       IF NOT : ERROR
 061C 96 05    B          LDAA    DIV3+1
 061E 27 07 0627          BEQ     CLRR1
 0620 81 63    A          CMPA    #99
 0622 2B 0C 0630          BMI     NOER
 0624 7E 06EE  P ER0      JMP     ER1
                  *
 0627 CE 0000  A CLRR1    LDX     #0
 062A FF 0070  D          STX     R1
 062D 7E 064C  P          JMP     OR1A
```

```
019   COMP   *** DATA COMPUTING ROUTINE ***

                     *
0630 CE 0000   A NOER   LDX    #0
0633 DF 02     B       STX    DIV2      DIVIDEND IN DIV2 (HI) & DI
0635 86 0A     A       LDAA   #$0A
0637 5F                CLRB             DIVISOR IN A & B = 10
0638 BD 09F1   P       JSR    DDIV      QUOT/10
                     *
063B 96 05     B       LDAA   DIV3+1    LO BYTE QUOT.
063D 48                ASLA
063E 48                ASLA
063F 48                ASLA
0640 48                ASLA
0641 B7 0070   D       STAA   R1        SAVE 1ST DIG IN UPPER
                     *                          4 BITS OF R1
                     *
0644 96 03     B       LDAA   DIV2+1    LO BYTE REM.
0646 BB 0070   D       ADDA   R1
0649 B7 0070   D       STAA   R1        REM IN LOWER 4 BITS
                     *
                     *       NOW FORM 2 BCD DIGITS AFTER DEC PT
                     *
064C FE 006E   D OR1A  LDX    OR1       GET REM.
064F DF 04     B       STX    MUL3
0651 CE 000A   A       LDX    #10       MULTIPLY BY 10
0654 DF 00     B       STX    MUL1
0656 BD 0997   P       JSR    MUL
                     *                    PRODUCT: DIV2 (HI) & DIV3 (LO)
0659 FE 0058   D       LDX    EX        GET OFFSET
065C A6 2B     B       LDAA   SAAXA+1,X LO BYTE DIVISOR
065E E6 2A     B       LDAB   SAAXA,X   HI
0660 BD 09F1   P       JSR    DDIV
                     *                     RETURNS: Q-DIV3 & R-DIV2
0663 96 05     B       LDAA   DIV3+1    LO BYTE
0665 48                ASLA
0666 48                ASLA
0667 48                ASLA
0668 48                ASLA             PACK DIGIT INTO UPPER
0669 B7 0072   D       STAA   R2        4 BITS OF R2
                     *
066C DE 02     B       LDX    DIV2      GET REMAINDER
066E DF 04     B       STX    MUL3
0670 CE 000A   A       LDX    #10       *10
0673 DF 00     B       STX    MUL1
0675 BD 0997   P       JSR    MUL
                     *
0678 FE 0058   D       LDX    EX
067B A6 2B     B       LDAA   SAAXA+1,X (REM*10)/(SAA-XA)
067D E6 2A     B       LDAB   SAAXA,X
067F BD 09F1   P       JSR    DDIV      2ND DIGIT AFTER DP
                     *
0682 96 05     B       LDAA   DIV3+1
0684 BB 0072   D       ADDA   R2        FIRST 2 DIGITS AFTER
0687 B7 0072   D       STAA   R2        DEC.PT. IN R2
                     *
                     *
                     *
                     *    CONVERT BCD # BACK TO BIN #
                     *
```

```
020   COMP   *** DATA COMPUTING ROUTINE ***

068A B6 0070  D        LDAA   R1        MSD
068D F6 0072  D        LDAB   R2        LSD
                *
0690 BD 0000  A        JSR    BCDBIN
                *
0693 CE 0000  A        LDX    #0
0696 DF 02    B        STX    MUL2
0698 B7 0070  D        STAA   R1        HI BYTE MULTIPLICAN
069B F7 0071  D        STAB   R1+1      LO BYTE
                *
069E FE 0058  D        LDX    EX
06A1 EE 00    A        LDX    STCN1H,X GET STD CON. VALUE
06A3 DF 00    B        STX    MUL1      MULTIPLIER (SAC)
                *
06A5 FE 0070  D        LDX    R1        GET MULTIPLICAN (XA/(SAA-X
06A8 DF 04    B        STX    MUL3
06AA BD 0997  P        JSR    MUL       // (XA/(SAA-XA))*SAC
                *
                *   NOW MUST DIVIDE ANSWER BY 100000 TO
                *     SCALE INTEGER VALUE FOR CORRECT
                *     DEC. PT. POSITION
                *   DIVIDEND IN DIV2 (HI) & DIV3 (LO)
                *
06AD 86 E8    A        LDAA   #$E8
06AF C6 03    A        LDAB   #$03      DIVISOR = 1000
06B1 BD 09F1  P        JSR    DDIV      /1000
                *
06B4 CE 0000  A        LDX    #0
06B7 DF 02    B        STX    DIV2      IGNORE REM
06B9 86 64    A        LDAA   #$64
06BB 5F                CLRB             DIVISOR = 100
06BC BD 09F1  P        JSR    DDIV      /100
                *
                *     DIV3 = QUOT.   &   DIV2 = REM
                *
                *
06BF B6 0033  D        LDAA   ELNUM
06C2 B7 0059  D        STAA   EX+1
06C5 7F 0058  D        CLR    EX
06C8 FE 0058  D        LDX    EX        BUFF. OFFSET
06CB 96 04    B        LDAA   DIV3
06CD A7 08    B        STAA   CONBF,X
06CF 96 05    B        LDAA   DIV3+1
06D1 A7 09    B        STAA   CONBF+1,X
                *
06D3 DE 02    B        LDX    DIV2      GET REM
06D5 DF 04    B        STX    DIV3      SET UP FOR DIV.
06D7 CE 0000  A        LDX    #0
06DA DF 02    B        STX    DIV2      CLEAR UPPER 16 BITS
06DC 86 0A    A        LDAA   #$0A
06DE 5F                CLRB             DIVISOR =  10
06DF BD 09F1  P        JSR    DDIV      /10
                *
06E2 FE 0058  D        LDX    EX
06E5 96 05    B        LDAA   DIV3+1    LO BYTE QUOT
06E7 A7 0A    B        STAA   CONBF+2,X
                *
06E9 96 03    B        LDAA   DIV2+1    STORE REM. TO CONBF+3
```

```
021   COMP   *** DATA COMPUTING ROUTINE ***

 06EB A7 0B     B        STAA   CONBF+3,X
                *
 06ED 39                 RTS
                *
                *
 06EE B6 0033   D ER1    LDAA   ELNUM     CURRENT ELEMENT
 06F1 27 0A 06FD         BEQ    Z1
                *
 06F3 81 04     A        CMPA   #4
 06F5 27 0A 0701         BEQ    CAD
                *
 06F7 81 08     A        CMPA   #8
 06F9 27 0A 0705         BEQ    LEAD
                *
 06FB 20 0C 0709         BRA    COP
                *
 06FD 7C 0076   D Z1     INC    Z1F
 0700 39                 RTS
                *
 0701 7C 0077   D CAD    INC    CADF
 0704 39                 RTS
                *
 0705 7C 0079   D LEAD   INC    LEADF
 0708 39                 RTS
                *
 0709 7C 0078   D COP    INC    COPF
 070C 39                 RTS
                *
                *PRINT CONCENTRATIONS FOR ONE SAMPLE ON THE
                *PRINTER
                *
 070D 7F 004D   D PRTSMP CLR    ONES
 0710 7F 004E   D        CLR    TENS
 0713 7F 004F   D        CLR    HUND
 0716 86 01     A        LDAA   #1
 0718 B7 0000   A        STAA   BLOCK
                *
 071B B6 0036   D        LDAA   SAMPLE    SAMPLE =0?
 071E 4D                 TSTA             YES: PRINT OUT HEADER MESS
 071F 26 0C 072D         BNE    SK30
 0721 CE 0000   A        LDX    #MES45
 0724 BD 0000   A        JSR    PRINT1
 0727 BD 0000   A        JSR    SUB3      LF
 072A BD 0000   A        JSR    SUB3
                *
 072D 7D 0000   A SK30   TST    ZINK
 0730 27 10 0742         BEQ    SK31
                *
 0732 7D 0076   D        TST    Z1F       ZN ERROR?
 0735 27 19 0750         BEQ    SK32      NO
 0737 CE 5A4E   A        LDX    #$5A4E
 073A FF 0019   A        STX    MESERR+25
 073D 7C 0075   D        INC    ERFLAG
 0740 20 0E 0750         BRA    SK32
                *
 0742 7D 0078   D SK31   TST    COPF      CU ERROR?
 0745 27 09 0750         BEQ    SK32      NO
 0747 CE 4355   A        LDX    #$4355    YES
```

```
022   COMP   *** DATA COMPUTING ROUTINE ***

 074A FF 0019  A          STX     MESERR+25
 074D 7C 0075  D          INC     ERFLAG
                  *
 0750 7D 0079  D SK32     TST     LEADF     PB ERROR?
 0753 27 09 075E          BEQ     SK33      NO
 0755 CE 5042  A          LDX     #$5042    YES
 0758 FF 001C  A          STX     MESERR+28
 075B 7C 0075  D          INC     ERFLAG
                  *
 075E 7D 0077  D SK33     TST     CADF      CD ERROR?
 0761 27 09 076C          BEQ     SK34      NO
 0763 CE 4344  A          LDX     #$4344    YES
 0766 FF 001F  A          STX     MESERR+31
 0769 7C 0075  D          INC     ERFLAG
                  *
 076C 7D 0075  D SK34     TST     ERFLAG
 076F 27 15 0786          BEQ     SK35
                  *
 0771 CE 0000  A          LDX     #MESERR   IF THERE WAS AN
 0774 BD 0000  A          JSR     PRINT1    ERROR, PRINT MESG
 0777 7F 0076  D          CLR     Z1F
 077A 7F 0078  D          CLR     COPF
 077D 7F 0077  D          CLR     CADF      CLEAR ALL FLAGS
 0780 7F 0079  D          CLR     LEADF
 0783 7F 0075  D          CLR     ERFLAG
                  *
 0786 BD 0796  P SK35     JSR     CLPRBF
 0789 BD 07A4  P          JSR     PRDATE    PRINT DATE LINE
 078C BD 0796  P          JSR     CLPRBF
 078F BD 081F  P          JSR     PRCONS    CONC. LINE
 0792 BD 0000  A          JSR     SUB3
 0795 39                  RTS
                  *
 0796 C6 28    A CLPRBF   LDAB    #40       CLEAR PRINT BUFFER
 0798 86 20    A          LDAA    #$20
 079A CE 0000  D          LDX     #PRBUF
 079D A7 00    A CLPR1    STAA    0,X
 079F 08                  INX
 07A0 5A                  DECB
 07A1 26 FA 079D          BNE     CLPR1
 07A3 39                  RTS
                  *
                  *
                  * PRINT THE SAMPLE NUMBER
                  *   AND THE DATE AND TIME
                  *
 07A4 CE 0000  D PRDATE   LDX     #PRBUF    /ADDRESS OF PRINT BUFFER
 07A7 86 2A    A          LDAA    #'*
 07A9 A7 00    A          STAA    0,X
 07AB 08                  INX
 07AC F6 0036  D          LDAB    SAMPLE    /SAMPLE #
 07AF 5C                  INCB               NO 0 SAMPLE #
 07B0 4F                  CLRA               /HIGH BYTE IS 0
 07B1 7C 004F  D          INC     HUND
 07B4 BD 0910  P          JSR     BINTOA    /CORRECT BINARY TO ASCII
 07B7 7F 004F  D          CLR     HUND
 07BA CE 0008  D          LDX     #PRBUF+8  /START OF DATA
 07BD F6 0000  A          LDAB    YRL       /YEAR LOW BYTE
```

```
023   COMP    *** DATA COMPUTING ROUTINE ***

07C0 B6 0000  A         LDAA    YRH       /YEAR HIGH BYTE
07C3 7C 0074  D         INC     ZESUP     DON'T ZERO SUPRES
07C6 BD 0910  P         JSR     BINTOA    /CONVERT & PUT IN BUF
07C9 08                 INX               LEAVE 2 BLANKS
07CA 08                 INX
07CB 7C 004F  D         INC     HUND
07CE F6 0000  A         LDAB    DAYL      /DAY OF THE YEAR
07D1 B6 0000  A         LDAA    DAYH
07D4 BD 0910  P         JSR     BINTOA    /CONVERT & PUT IN BUF
07D7 7F 004F  D         CLR     HUND
07DA 08                 INX               /2 BLANKS
07DB 08                 INX
07DC FF 0094  D         STX     SAVX      SAVE CURRENT X
07DF FE 0092  D         LDX     TMBUF     GET SAMPLE AQUIS. TIME
07E2 A6 00    A         LDAA    0,X
07E4 08                 INX
07E5 E6 00    A         LDAB    0,X
07E7 08                 INX
07E8 FF 0092  D         STX     TMBUF     UPDATE POINTER
07EB FE 0094  D         LDX     SAVX      RESET X
                *
07EE BD 0000  A         JSR     BCDBIN    CONV. BCD TO BIN
                *
07F1 BD 0910  P         JSR     BINTOA    /CONVERT & PUT IN BUF
                *
07F4 86 3A    A         LDAA    #':
07F6 A7 00    A         STAA    0,X
07F8 08                 INX
07F9 4F                 CLRA
07FA FF 0094  D         STX     SAVX
07FD FE 0092  D         LDX     TMBUF
0800 E6 00    A         LDAB    0,X
0802 08                 INX
0803 FF 0092  D         STX     TMBUF
0806 FE 0094  D         LDX     SAVX
0809 7C 004E  D         INC     TENS
080C BD 0000  A         JSR     BCDBIN
080F BD 0910  P         JSR     BINTOA
0812 7F 004E  D         CLR     TENS
                *
0815 CE 0000  D         LDX     #PRBUF
0818 BD 0000  A         JSR     PRINT1
081B 7F 0074  D         CLR     ZESUP
081E 39                 RTS
                *
                *PRINT THE CONCENTRATION LINE
                *
081F 7D 0000  A PRCONS  TST     ZINK      LOOKING FOR ZINC?
0822 27 3A 085E         BEQ     PNT1      NO: SKIP ZN
                *
0824 CE 0005  D         LDX     #PRBUF+5 FILL BUFFER
0827 86 5A    A         LDAA    #'Z
0829 A7 00    A         STAA    0,X
082B 08                 INX
082C 86 4E    A         LDAA    #'N
082E A7 00    A         STAA    0,X
0830 08                 INX
0831 86 3A    A         LDAA    #':
```

```
024   COMP   *** DATA COMPUTING ROUTINE ***

0833 A7 00     A          STAA   0,X
0835 08                   INX
                  *
0836 7C 004F   D          INC    HUND      # <= 999
0839 96 08     B          LDAA   CONBF     ZN CONC. VAL
083B D6 09     B          LDAB   CONBF+1
                  *
083D BD 0910   P          JSR    BINTOA
0840 BD 090A   P          JSR    DECPT
                  *
0843 D6 0A     B          LDAB   CONBF+2   # AFTER DEC PT
0845 4F                   CLRA
0846 7C 004D   D          INC    ONES
0849 7C 0074   D          INC    ZESUP     DON'T SUPRESS 0'S
                  *
084C BD 0910   P          JSR    BINTOA
                  *
084F D6 0B     B          LDAB   CONBF+3
0851 4F                   CLRA
0852 BD 0910   P          JSR    BINTOA
0855 7F 004D   D          CLR    ONES
0858 7F 0074   D          CLR    ZESUP
085B 7E 0895   P          JMP    PNT2
                  *
085E CE 0005   D PNT1     LDX    #PRBUF+5
0861 86 43     A          LDAA   #'C
0863 A7 00     A          STAA   0,X
0865 08                   INX
0866 86 55     A          LDAA   #'U
0868 A7 00     A          STAA   0,X
086A 08                   INX
086B 86 3A     A          LDAA   #':
086D A7 00     A          STAA   0,X
086F 08                   INX
                  *
0870 7C 004F   D          INC    HUND
0873 96 14     B          LDAA   CONBF+12  CU
0875 D6 15     B          LDAB   CONBF+13
0877 BD 0910   P          JSR    BINTOA
087A BD 090A   P          JSR    DECPT
                  *
087D D6 16     B          LDAB   CONBF+14
087F 4F                   CLRA
0880 7C 004D   D          INC    ONES
0883 7C 0074   D          INC    ZESUP
0886 BD 0910   P          JSR    BINTOA
                  *
0889 D6 17     B          LDAB   CONBF+15
088B 4F                   CLRA
088C BD 0910   P          JSR    BINTOA
088F 7F 004D   D          CLR    ONES
0892 7F 0074   D          CLR    ZESUP
                  *
0895 CE 0011   D PNT2     LDX    #PRBUF+17
0898 86 50     A          LDAA   #'P
089A A7 00     A          STAA   0,X
089C 08                   INX
089D 86 42     A          LDAA   #'B
```

```
025    COMP    *** DATA COMPUTING ROUTINE ***

089F A7 00     A        STAA   0,X
08A1 08                 INX
08A2 86 3A     A        LDAA   #':
08A4 A7 00     A        STAA   0,X
08A6 08                 INX
                  *
08A7 96 10     B        LDAA   CONBF+8
08A9 D6 11     B        LDAB   CONBF+9
08AB BD 0910   P        JSR    BINTOA
08AE BD 090A   P        JSR    DECPT
                  *
08B1 D6 12     B        LDAB   CONBF+10
08B3 4F                 CLRA
08B4 7C 004D   D        INC    ONES
08B7 7C 0074   D        INC    ZESUP
08BA BD 0910   P        JSR    BINTOA
                  *
08BD D6 13     B        LDAB   CONBF+11
08BF 4F                 CLRA
08C0 BD 0910   P        JSR    BINTOA
08C3 7F 004D   D        CLR    ONES
08C6 7F 0074   D        CLR    ZESUP
                  *
08C9 CE 001D   D        LDX    #PRBUF+29
08CC 86 43     A        LDAA   #'C
08CE A7 00     A        STAA   0,X
08D0 08                 INX
08D1 86 44     A        LDAA   #'D
08D3 A7 00     A        STAA   0,X
08D5 08                 INX
08D6 86 3A     A        LDAA   #':
08D8 A7 00     A        STAA   0,X
08DA 08                 INX
                  *
08DB 96 0C     B        LDAA   CONBF+4   CD
08DD D6 0D     B        LDAB   CONBF+5
08DF BD 0910   P        JSR    BINTOA
08E2 BD 090A   P        JSR    DECPT
08E5 7F 004F   D        CLR    HUND
                  *
08E8 D6 0E     B        LDAB   CONBF+6
08EA 4F                 CLRA
08EB 7C 004D   D        INC    ONES
08EE 7C 0074   D        INC    ZESUP
08F1 BD 0910   P        JSR    BINTOA
                  *
08F4 D6 0F     B        LDAB   CONBF+7
08F6 4F                 CLRA
08F7 BD 0910   P        JSR    BINTOA
08FA 7F 004D   D        CLR    ONES
08FD 7F 0074   D        CLR    ZESUP
                  *
                  *
0900 CE 0000   D        LDX    #PRBUF
0903 BD 0000   A        JSR    PRINT1
0906 BD 0000   A        JSR    SUB3
0909 39                 RTS
                  *
```

```
026   COMP    *** DATA COMPUTING ROUTINE ***

                      *
                      *
                      * PUT A DECIMAL POINT IN THE CONCENTRATION
                      *
090A 86 2E     A DECPT  LDAA   #'.
090C A7 00     A        STAA   0,X
090E 08                 INX
090F 39                 RTS
                      *
                      *
                      *CONVERT THE NUMBER IN A AND B TO ASCII
                      *AND PUT IT IN THE MEMORY THAT IS POINTED TO BY X
         0004   B DVQUO  EQU    MUL3
         0002   B DVREM  EQU    MUL2
0910 FF 002C   D BINTOA STX    BINTP1
0913 B7 002E   D        STAA   BINTP2
0916 F7 002F   D        STAB   BINTP2+1
0919 FE 002E   D        LDX    BINTP2
091C 86 0A     A        LDAA   #10
091E 5F                 CLRB
091F BD 09EF   P        JSR    DIV       /DIVIDE BY 10
0922 96 03     B        LDAA   DVREM+1   /REMAINDER
0924 B7 0061   D        STAA   BIONES    /ONES DIGIT
                      *
0927 B6 004D   D        LDAA   ONES
092A 4D                 TSTA
092B 27 06 0933         BEQ    SK20
092D FE 002C   D        LDX    BINTP1
0930 7E 0981   P        JMP    ONE1
                      *
0933 DE 04     B SK20   LDX    DVQUO
0935 86 0A     A        LDAA   #10
0937 5F                 CLRB
0938 BD 09EF   P        JSR    DIV
093B 96 03     B        LDAA   DVREM+1   /REMAINDER
093D B7 0060   D        STAA   BITENS    /TENS DIGIT
                      *
0940 B6 004E   D        LDAA   TENS
0943 4D                 TSTA
0944 27 06 094C         BEQ    SK21
0946 FE 002C   D        LDX    BINTP1
0949 7E 097C   P        JMP    TEN1
                      *
094C DE 04     B SK21   LDX    DVQUO
094E 86 0A     A        LDAA   #10
0950 5F                 CLRB
0951 BD 09EF   P        JSR    DIV
0954 96 03     B        LDAA   DVREM+1
0956 B7 005F   D        STAA   BIHUND    /HUNDREDS DIGIT
                      *
0959 B6 004F   D        LDAA   HUND
095C 4D                 TSTA
095D 27 06 0965         BEQ    SK22
095F FE 002C   D        LDX    BINTP1
0962 7E 0977   P        JMP    HUN1
                      *
0965 DE 04     B SK22   LDX    DVQUO
0967 86 0A     A        LDAA   #10
```

```
027   COMP    *** DATA COMPUTING ROUTINE ***

 0969 5F                    CLRB
 096A BD 09EF  P            JSR     DIV
 096D 96 03    B            LDAA    DVREM+1
 096F B7 005E  D            STAA    BITHOU    /THOUSANDS DIGIT
 0972 FE 002C  D            LDX     BINTP1
 0975 8D 10 0987            BSR     CVTDGT
 0977 B6 005F  D HUN1       LDAA    BIHUND
 097A 8D 0B 0987            BSR     CVTDGT
 097C B6 0060  D TEN1       LDAA    BITENS
 097F 8D 06 0987            BSR     CVTDGT
 0981 B6 0061  D ONE1       LDAA    BIONES
 0984 8D 01 0987            BSR     CVTDGT    /ALWAYS CONVERT LAST DIGIT
 0986 39                    RTS
                  *
 0987 7D 0074  D CVTDGT     TST     ZESUP     IF SET: DON'T
 098A 26 03 098F            BNE     CVTDG2    SUPRESS 0'S
 098C 4D                    TSTA              ZERO SUPRESION
 098D 27 06 0995            BEQ     CVTDG1
 098F 8B 30    A CVTDG2     ADDA    #$30      / MAKE ASCII #
 0991 A7 00    A            STAA    0,X
 0993 08                    INX
 0994 39                    RTS
 0995 08          CVTDG1    INX
 0996 39                    RTS
                  *
                  *
                  *MULTIPLY TWO 16 BIT SIGNED NUMBERS YIELDING A 32
                  *NUMBER. CALL WITH*
                  *MULTIPLIER IN MUL1
                  *MULTIPLICAND IN MUL3
                  *RETURN WITH:
                  *PRODUCT HIGH 16 BITS IN MUL2
                  *PRODUCT LOW 16 BITS IN MUL3
                  *
                  *
                  *
 0997 7F 0062  D MUL        CLR     RETF
 099A DE 04    B            LDX     MUL3
 099C DF 06    B            STX     MUL4
 099E CE 0004  A            LDX     #4
 09A1 4F                    CLRA
 09A2 A7 01    B LP1        STAA    MUL1+1,X
 09A4 09                    DEX
 09A5 26 FB 09A2            BNE     LP1       //CLEAR WORKING REGISTERS
 09A7 CE 0010  A            LDX     #16       //SET SHIFT COUNT TO 16
 09AA 96 01    B LP2        LDAA    MUL1+1
 09AC 84 01    A            ANDA    #1        //GET Y(LSBIT)
 09AE 16                    TAB               //SAVE Y(LSBIT) IN ACCB
 09AF B8 0062  D            EORA    RETF      //Y(LSBIT)=Y(LSBIT-1) ?
 09B2 27 1D 09D1            BEQ     SHIFT     //YES: GO TO SHIFT ROUTINE
 09B4 5D                    TSTB              //NO: DOES Y(LSBIT) = 0 ?
 09B5 27 0E 09C5            BEQ     ADD       //YES: GO TO ADD ROUTINE
 09B7 96 03    B            LDAA    MUL2+1    //NO: SUBTRACT MULTIPLICAN
 09B9 D6 02    B            LDAB    MUL2      //FROM THE PRODUCT WITH TH
 09BB 90 07    B            SUBA    MUL4+1    //MSBYTES LINED UP,
 09BD D2 06    B            SBCB    MUL4
 09BF 97 03    B            STAA    MUL2+1
 09C1 D7 02    B            STAB    MUL2
```

```
028   COMP   *** DATA COMPUTING ROUTINE ***

09C3 20 0C 09D1          BRA    SHIFT     //THEN GO TO SHIFT ROUTINE
09C5 96 03     B ADD     LDAA   MUL2+1    //ADD MULITPLICAND TO THE
09C7 D6 02     B         LDAB   MUL2      //PRODUCT WITH THE MSBYTES
09C9 9B 07     B         ADDA   MUL4+1    //LINED UP.
09CB D9 06     B         ADCB   MUL4
09CD 97 03     B         STAA   MUL2+1
09CF D7 02     B         STAB   MUL2
09D1 7F 0062   D SHIFT   CLR    RETF      //CLEAR TEST BYTE
09D4 76 0000   B         ROR    MUL1      //SHIFT THE MULTIPLIER RIG
09D7 76 0001   B         ROR    MUL1+1    //ONE BIT SHIFTING THE LSP
09DA 79 0062   D         ROL    RETF      //INTO THE LSBIT OF FF.
09DD 77 0002   B         ASR    MUL2      //SHIFT THE PRODUCT RIGHT
09E0 76 0003   B         ROR    MUL2+1    //BIT, THE MSB REMAINING T
09E3 76 0004   B         ROR    MUL3      //SAME.
09E6 76 0005   B         ROR    MUL3+1
09E9 09                  DEX               //DECREMENT THE SHIFT COUN
09EA 26 BE 09AA          BNE    LP2       //IF NOT 0 CONTINUE
09EC DE 04     B         LDX    MUL3      //RETURN PRODUCT LOWER 16
09EE 39                  RTS

                *
                *DIV-CALL WITH:
                *16 BIT DIVIDEND IN X
                *16 BIT DIVISOR IN A(LOW) AND B(HIGH)
                *
                * DDIV - DOUBLE DIVIDE - CALL WITH:
                *   32 BIT DIVIDEND IN DIV2 AND DIV3
                *   DIVISOR LOW IN A REG
                *   DIVISOR HI IN B REG
                *
                *
                *RETURNS:
                *  16 BIT QUOTIENT IN DIV3
                *  16 BIT REMAINDER IN DIV2
                ****************************
09EF 8D 63 0A54 DIV      BSR    STOPDS    //DIVIDEND LOW
09F1 7F 0062   D DDIV    CLR    RETF      //RETURN QUOTIENT
09F4 97 01     B LD9     STAA   DIV1+1    //DIVISOR LOW
09F6 D7 00     B         STAB   DIV1      //DIVISOR HIGH
09F8 7F 0063   D         CLR    SIGN      //FLAG FOR SIGN OF RESULTS
09FB 7D 0000   B         TST    DIV1      //DIVISOR SIGN
09FE 2A 07 0A07          BPL    LD1       //ITS POSITIVE
0A00 CE 0001   B         LDX    #DIV1+1   //LOW BYTE OF DIVISOR
0A03 C6 02     A         LDAB   #2        //TWO BYTES LONG
0A05 8D 69 0A70          BSR    COMP      //MAKE IT POSITIVE
0A07 7D 0002   B LD1     TST    DIV2      //SIGN OF DIVIDEND
0A0A 2A 04 0A10          BPL    L5
0A0C C6 04     A         LDAB   #4        //4 BYTES LONG
0A0E 8D 5D 0A6D          BSR    COMPF     //MAKE IT POSITIVE
0A10 BD 0A8F   P L5      JSR    OVFTST    //TEST FOR OVERFLOW
0A13 CE 0011   A         LDX    #17       //# TIMES THRU MAIN LOOP
0A16 5F          L0      CLRB             //QUOTIENT BIT IN B
0A17 96 00     B         LDAA   DIV1      //DIVISOR UPPER
0A19 91 02     B         CMPA   DIV2      //DIVIDEND HIGH UPPER
0A1B 27 62 0A7F          BEQ    L2        //EQUAL-TEST LOWER BYTE
0A1D 22 0D 0A2C          BHI    L1        //DIVISOR > DIVIDEND
0A1F 5C          L3      INCB             //DIVISOR <= DIVIDEND
0A20 96 03     B         LDAA   DIV2+1    //DIVIDEND HIGH LOWER
0A22 90 01     B         SUBA   DIV1+1    //DIVISOR LOWER
```

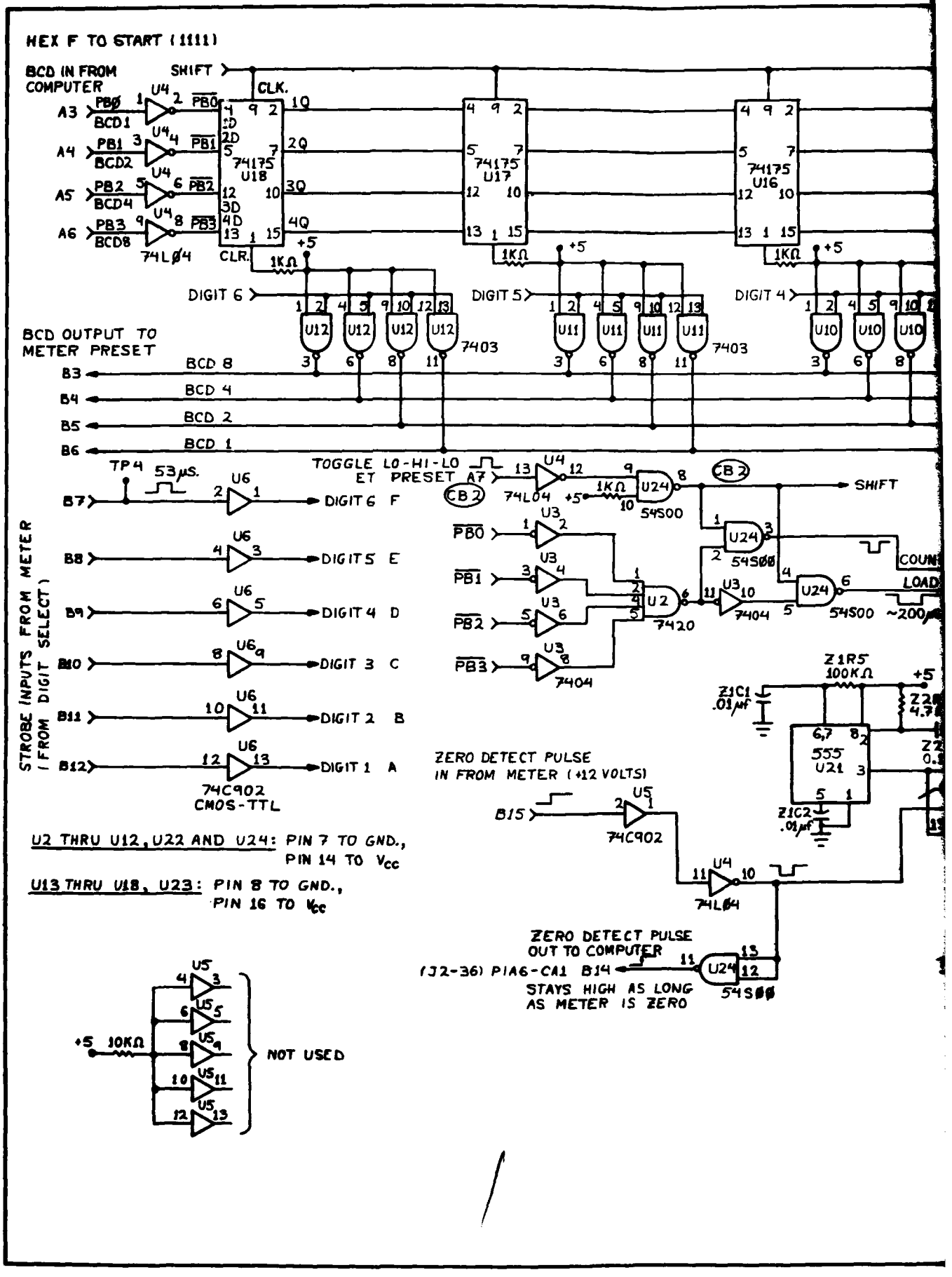
```
029   COMP   *** DATA COMPUTING ROUTINE ***

0A24 97 03     B        STAA   DIV2+1
0A26 96 02     B        LDAA   DIV2     //DIVIDEND HIGH UPPFR
0A28 92 00     B        SBCA   DIV1     //DIVISOR UPPER
0A2A 97 02     B        STAA   DIV2
0A2C 8C 0001   A L1      CPX    #1       //LAST TIME THRU
0A2F 27 56 0A87          BEQ    L7
0A31 8D 6F 0AA2          BSR    STSHF    //SHIFT DIVIDEND LEFT ONE
0A33 79 0003   B         ROL    DIV2+1
0A36 79 0002   B         ROL    DIV2
0A39 DA 05     B L6      ORAB   DIV3+1   //PUT IN QUOTIENT BIT
0A3B D7 05     B         STAB   DIV3+1
0A3D 09                  DEX
0A3E 26 D6 0A16          BNE    L0       //DO LOOP 17 TIMES
0A40 86 01     A         LDAA   #1
0A42 B4 0063   D         ANDA   SIGN     //MAY NEED TO COMPLEMENT R
0A45 27 04 0A4B          BEQ    L8
0A47 C6 02     A         LDAB   #2
0A49 8D 22 0A6D          BSR    COMPF    //COMPLEMENT QUOTIENT
0A4B 7D 0062   D L8      TST    RETF     //RETURN FLAG
0A4E 27 3B 0A8B          BEQ    L4       //RETURN QUOTIENT
0A50 DE 02     B         LDX    DIV2     //RETURN REMAINDER
0A52 0C                  CLC             //NO OVERFLOW
0A53 39                  RTS
0A54 DF 04     B STOPDS  STX    DIV3     //DIVIDEND LOW
0A56 CE 0000   A         LDX    #0
0A59 7D 0004   B         TST    DIV3     //SIGN BIT
0A5C 2A 01 0A5F          BPL    STOP1    //ITS POSITIVE
0A5E 09                  DEX             //UPPER WORD IS ALL ONES
0A5F DF 02     B STOP1   STX    DIV2     //DIVIDEND HIGH
0A61 39                  RTS
0A62 8D F0 0A54 REM      BSR    STOPDS   //STORE OPERAND IN X
0A64 CE FFFF   A         LDX    #$FFFF
0A67 FF 0062   D         STX    RETF     //RETURN REMAINDER
0A6A 7E 09F4   P         JMP    LD9
0A6D CE 0005   B COMPF   LDX    #DIV3+1  //DIVIDEND/QUOTIENT LOWER
0A70 0C          COMP    CLC             //COMPLEMENT NUM POINTED T
0A71 86 00     A CML1    LDAA   #0
0A73 A2 00     A         SBCA   0,X      //COMPLEMENT BYTE
0A75 A7 00     A         STAA   0,X      //STORE IT BACK
0A77 09                  DEX
0A78 5A                  DECB            //BYTE COUNTER
0A79 26 F6 0A71          BNE    CML1
0A7B 7C 0063   D         INC    SIGN     //CHANGED ONE ADDRESS
0A7E 39                  RTS
0A7F 96 01     B L2      LDAA   DIV1+1   //DIVISOR LOWER
0A81 91 03     B         CMPA   DIV2+1   //DIVIDEND HIGH LOWER
0A83 22 A7 0A2C          BHI    L1       //QUOTIENT BIT IS 0
0A85 20 98 0A1F          BRA    L3       //QUOTIENT BIT IS 1
0A87 8D 19 0AA2 L7       BSR    STSHF    //SHIFT ONLY QUOTIENT LAST
0A89 20 AE 0A39          BRA    L6
0A8B DE 04     B L4      LDX    DIV3     //QOTIENT
0A8D 0C                  CLC             //NO OVERFLOW
0A8E 39                  RTS
0A8F 96 02     B OVFTST  LDAA   DIV2     //DIVIDEND HIGH UPPER
0A91 91 00     B         CMPA   DIV1     //DIVISOR UPPER
0A93 2E 09 0A9E          BGT.   OVFYES
0A95 2D 06 0A9D          BLT    OVFNO
0A97 96 01     B         LDAA   DIV1+1   //DIVISOR LOWER
```

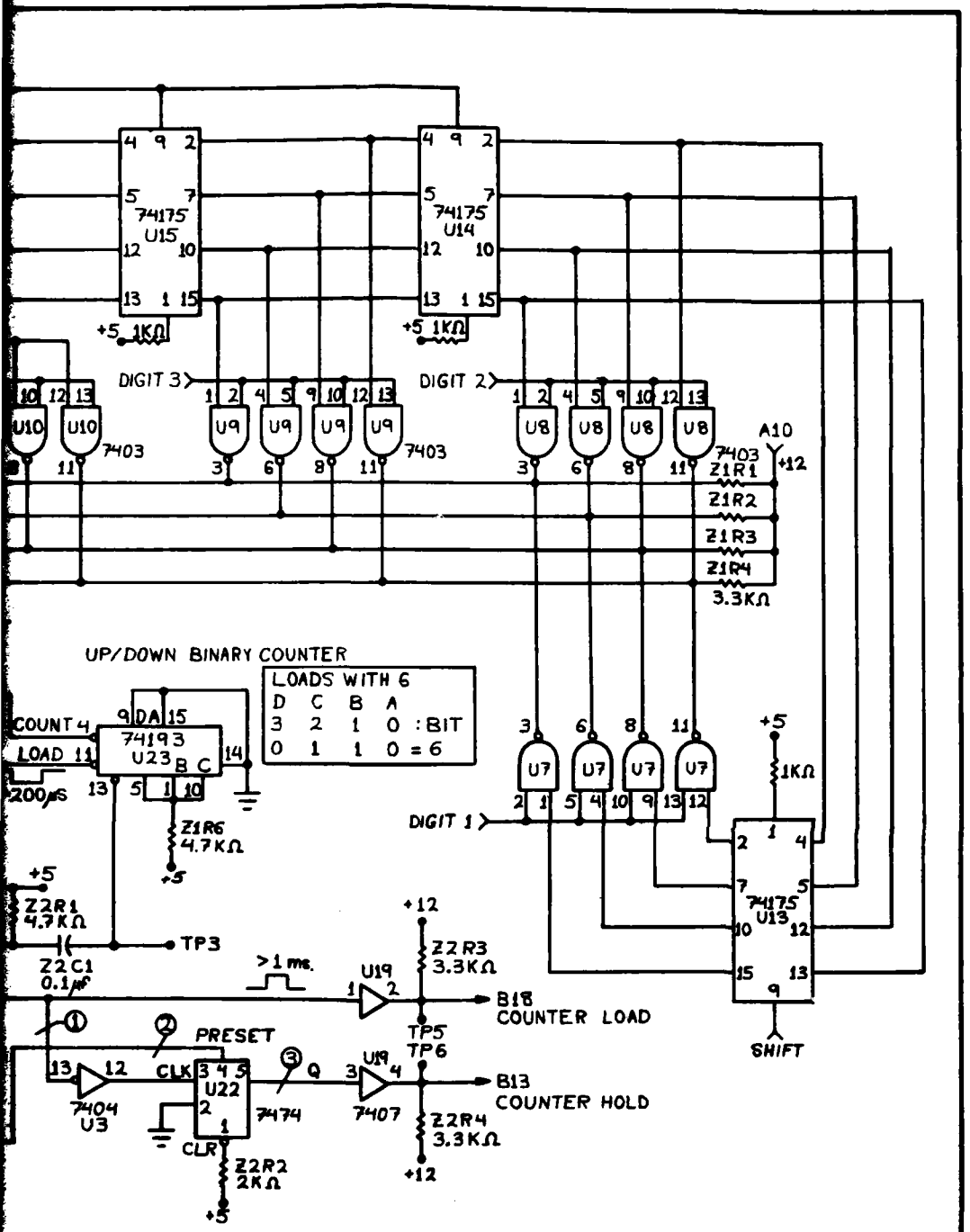
```
030   COMP   *** DATA COMPUTING ROUTINE ***

 0A99 91 03     B          CMPA   DIV2+1    //DIVIDEND HIGH LOWER
 0A9B 23 01 0A9E           BLS    OVFYES
 0A9D 39           OVFNO   RTS
 0A9E 32           OVFYES  PULA              //REMOVE OUR RETURN ADDR
 0A9F 32                   PULA
 0AA0 0D                   SEC               //SET OVERFLOW FLAG
 0AA1 39                   RTS               //RETURN TO DIV CALLER
 0AA2 78 0005   B  STSHF   ASL    DIV3+1
 0AA5 79 0004   B          ROL    DIV3
 0AA8 39                   RTS
                  * 16 BIT SIGNED COMPARE ROUTINE
                  *  COMPARE X TO A (HGIH) AND B (LOW)
                  *  RETURN (IN B):
                  *    -1 IF X < AB
                  *     0 IF X = AB
                  *     1 IF X > AB
                  *
 0AA9 FF 0030   D  CMP16   STX    TPC16     //TEMP
 0AAC B1 0030   D          CMPA   TPC16     //COMPARE HIGH BYTE
 0AAF 2D 0B 0ABC           BLT    CMP1      // X > AB
 0AB1 2E 0C 0ABF           BGT    CMM1      // X < AB
 0AB3 F1 0031   D          CMPB   TPC16+1   //COMPARE LOW BYTE
 0AB6 22 07 0ABF           BHI    CMM1      // X < AB
 0AB8 25 02 0ABC           BCS    CMP1      // X > AB
 0ABA 5F                   CLRB             // X = AB
 0ABB 39                   RTS
 0ABC C6 01     A  CMP1    LDAB   #1        // X > AB
 0ABE 39                   RTS
 0ABF C6 FF     A  CMM1    LDAB   #$FF      // X < AB
 0AC1 39                   RTS
                  *
                  *
                  *
                           XREF   VPRECY,CNTR2,DAYH,DAYL,HR,MIN
                           XREF   SEC,CNTR1,DATA,YRH,YRL,BCDBIN
                           XREF   STCN1H,BLOCK,PRINT1,SUB3,ZINK
                           XREF   MES45,MESERR,TDATA
                  *
                           XDEF   COMPT,LOWECO,HIECO,AREA,SAMPLE
                           XDEF   CONBF,MUL1,ERFLAG,TIMBUF,TMBUF
                           XDEF   PRBUF,FPTNO,LPTNO,CLSABF
                  *
                           END
ERRORS 00000
```

INDEX TO APPENDIX B

SYSTEM SCHEMATICS

Drawing No.	Description	Page
CC-1	Computer-to-Elapsed Timer Interface Board	139
CC-2A&B	LED Display of Valve Position Circuits	141
CC-3A&B	Valve Manual Override Interface	145
CC-4	Computer-to-Valve Relay Interface	149
CC-5	Low-Level-Sensor Circuit	151
CC-6	High-Low Sensor Display Circuits	153
CC-7	Fill-Empty Pump Circuit	155
CC-8	Main Pump Controller Circuit	157
CC-9	Pump Tach Circuit	159
CC-10	Valve Relay Circuit	161
CC-11A&B	Real-Time Clock and Keyboard Interface	163
CC-12	Real-Time Clock Circuit and LED Display Interface	167
CC-13	Elapsed Timer, Temperature, and Depth Meters Interface	169
CC-14	P.A.R. Potentiostat Back Plane and Front Panel	171
CC-15	Front Panel I/O (Switches)	173
CC-16	Front Panel I/O (Displays)	175
CC-17	Valve Manual Override – Front Panel I/O	177
CC-18	EPA-6800 CPU Board	179
CC-19	EPA-48 RAM/EPROM Board	181

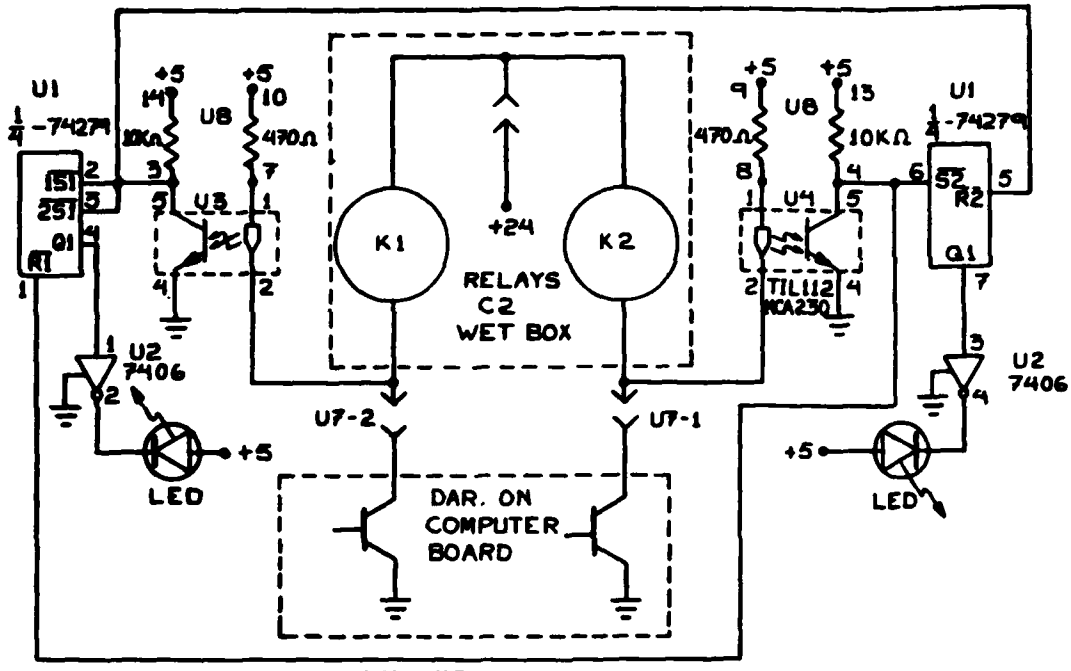

HEX F TO START (1111)
BCD IN FROM COMPUTER
SHIFT
CLK.
A3
A4
A5
A6
PBØ
PB1
PB2
PB3
BCD1
BCD2
BCD4
BCD8
74LØ4
74175
U18
U17
U16
CLR.
1KΩ
+5
DIGIT 6
DIGIT 5
DIGIT 4
7403
BCD OUTPUT TO METER PRESET
BCD 8
BCD 4
BCD 2
BCD 1
B3
B4
B5
B6
TOGGLE LO-HI-LO ET PRESET A7
74L04
54S00
CB2
SHIFT
TP4
53μS.
B7
B8
B9
B10
B11
B12
DIGIT 6 F
DIGIT 5 E
DIGIT 4 D
DIGIT 3 C
DIGIT 2 B
DIGIT 1 A
74C902
CMOS-TTL
STROBE INPUTS FROM METER (FROM DIGIT SELECT)
U2 THRU U12, U22 AND U24: PIN 7 TO GND., PIN 14 TO Vcc
U13 THRU U18, U23: PIN 8 TO GND., PIN 16 TO Vcc
7420
7404
54S08
LOAD
Z1R5
100KΩ
Z1C1
.01μf
555
U21
Z1C2
ZERO DETECT PULSE IN FROM METER (+12 VOLTS)
B15
ZERO DETECT PULSE OUT TO COMPUTER
(J2-36) PIA6-CA1 B14
STAYS HIGH AS LONG AS METER IS ZERO
10KΩ
NOT USED



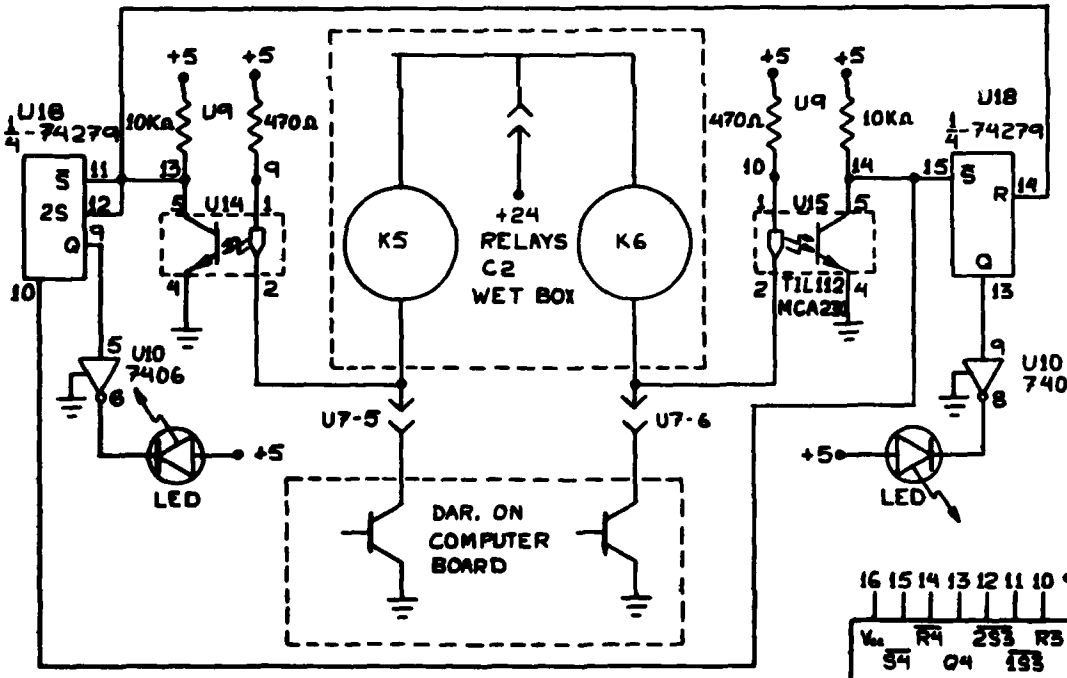
TO USE, ENTER:

1) ϕF_0
2) TOGGLE SHIFT (CB 2)
3) DUMMY DIGIT
4) TOGGLE SHIFT (TS)
5) 1ST DIGIT
   TS
6) 2ND DIGIT
   TS
7) 3RD DIGIT
   TS
8) 4TH DIGIT
   TS
9) 5TH DIGIT
   TS
10) 6TH DIGIT
    TS; WILL LOAD & START COUNTING

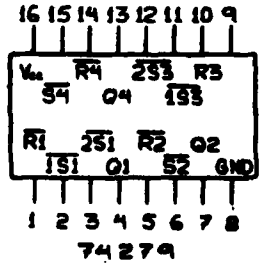
DESIGN C. CLAVELL	NOSC	DRAFT JMF
DRAWING NO. CC-1	2-28-78	REV. 3
LOGIC DIAGRAM - SEAWATER ANAL. TIMER INTERFACE TO ELAPSED TIMER		SHEET 1 OF 1

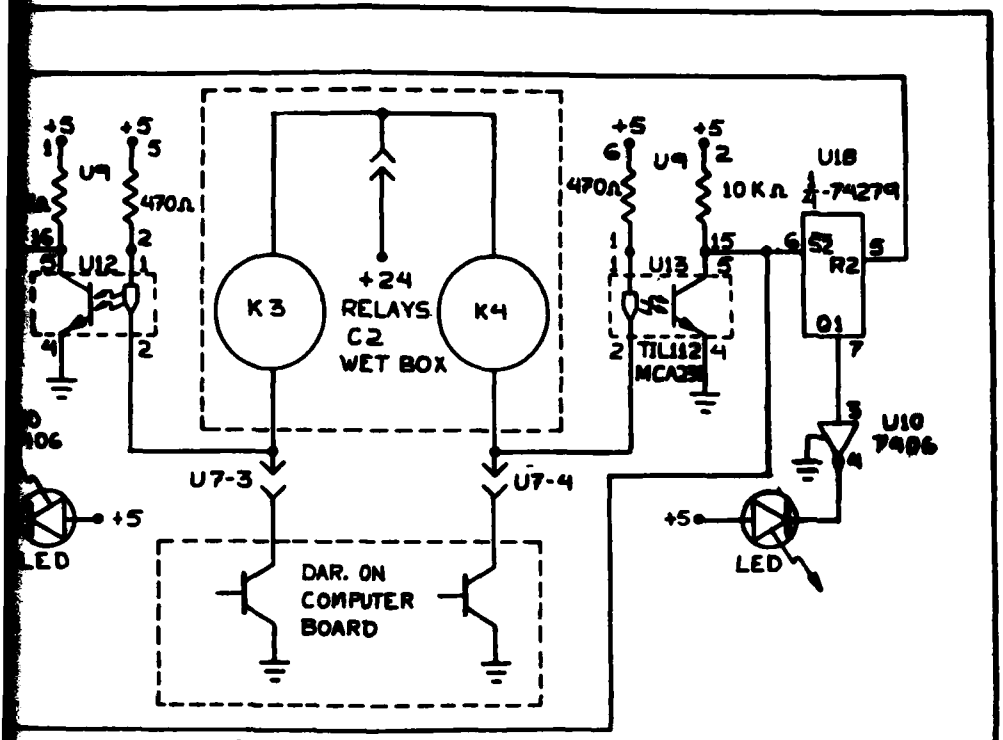


V1A - V1B

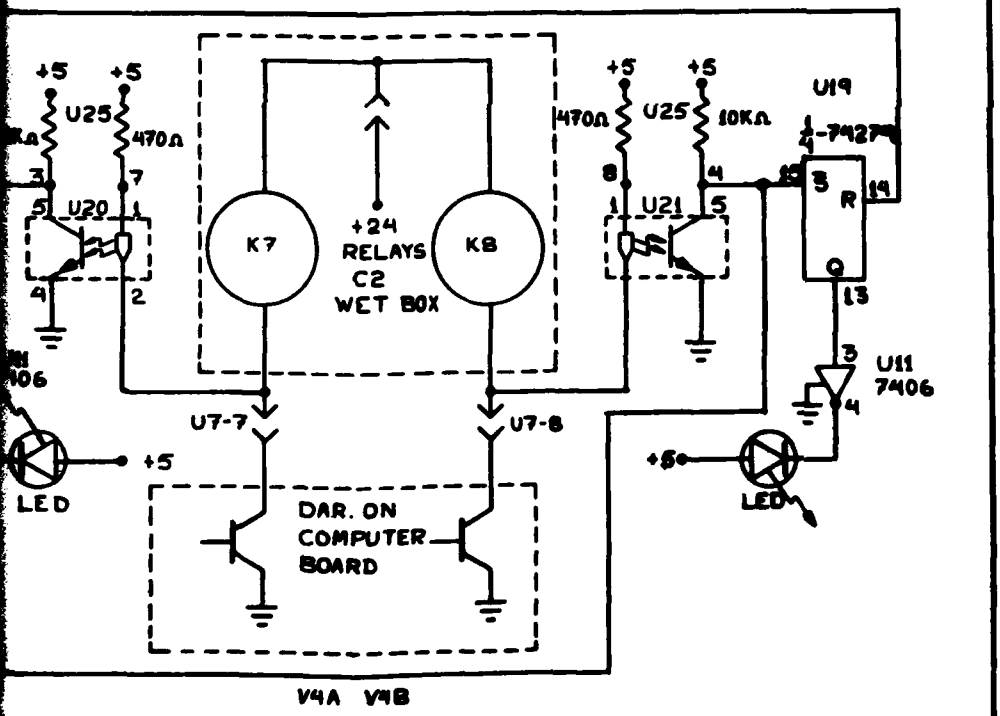


V3A V3B



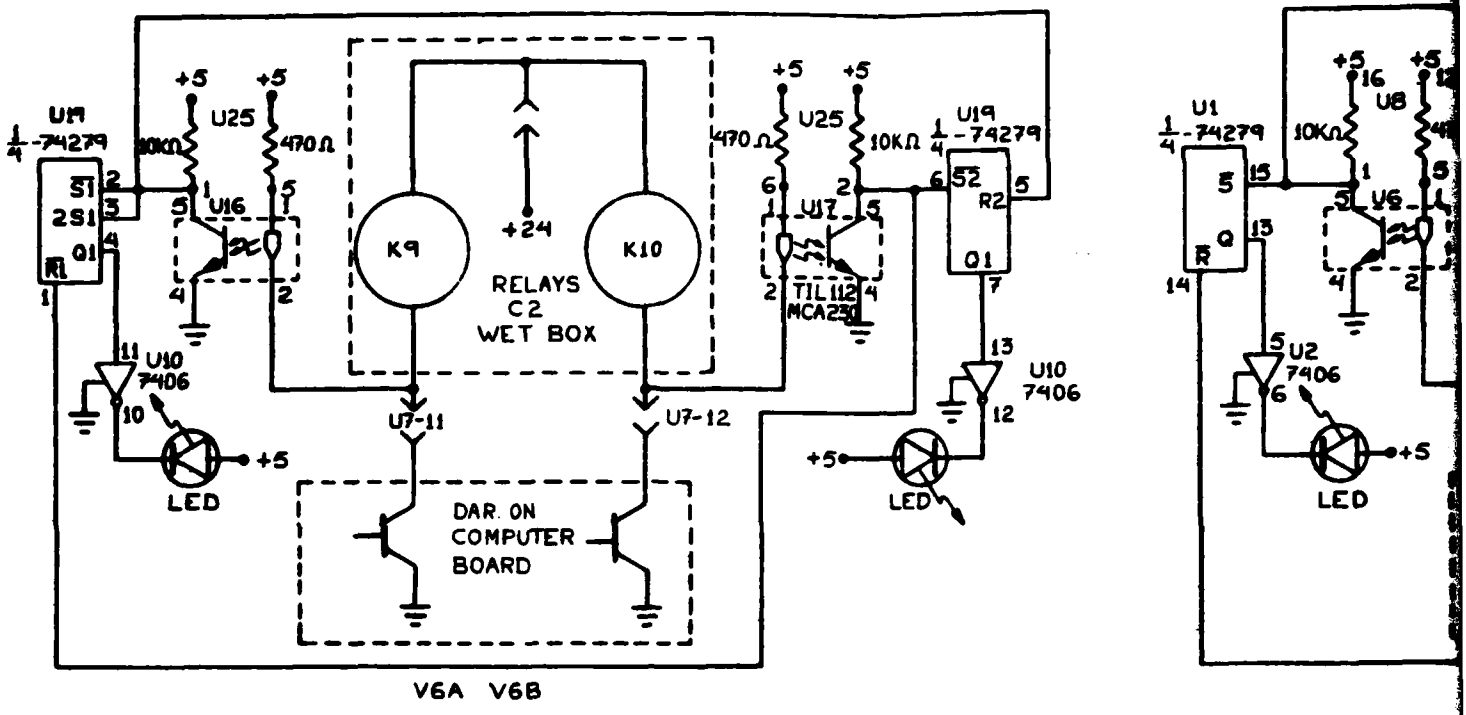


V2A V2B

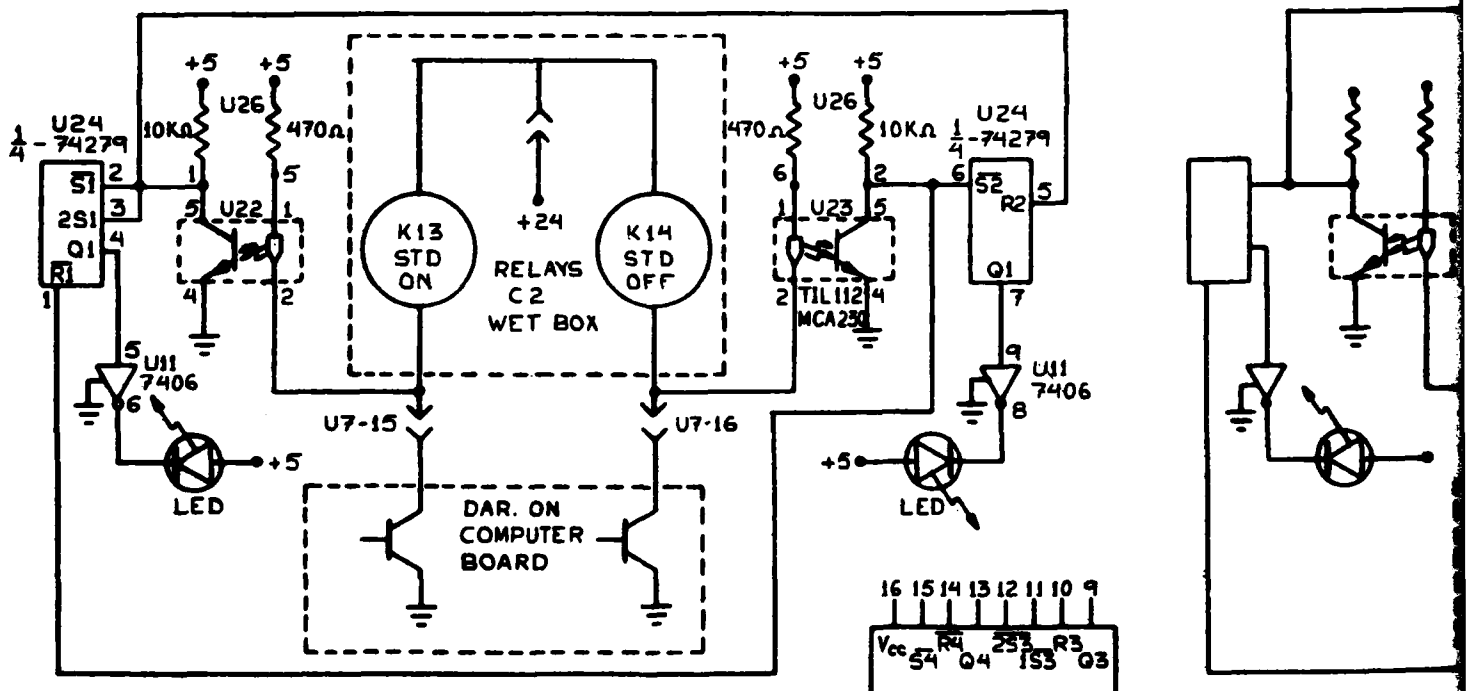


V4A V4B

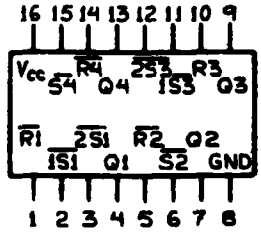
DESIGN C. CLARKE	NOSC	DRAFT JMF
DRW# CC-2A	9-7-79	REV# 1
LED DISPLAY OF VALVE POSITION		SHT 1 OF 2



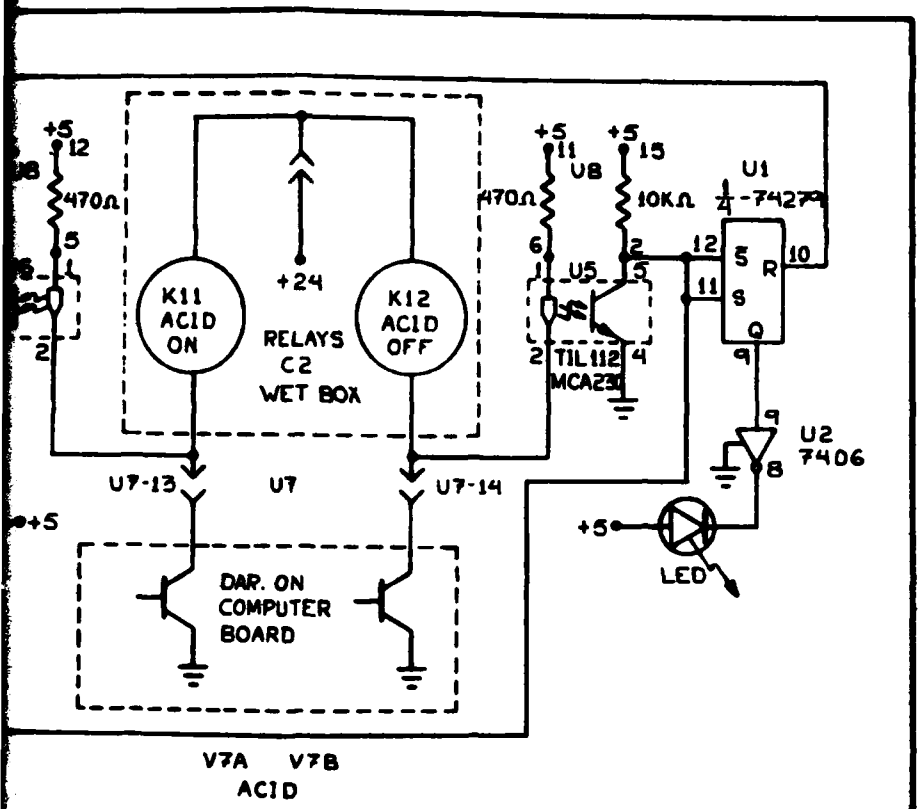
V6A V6B

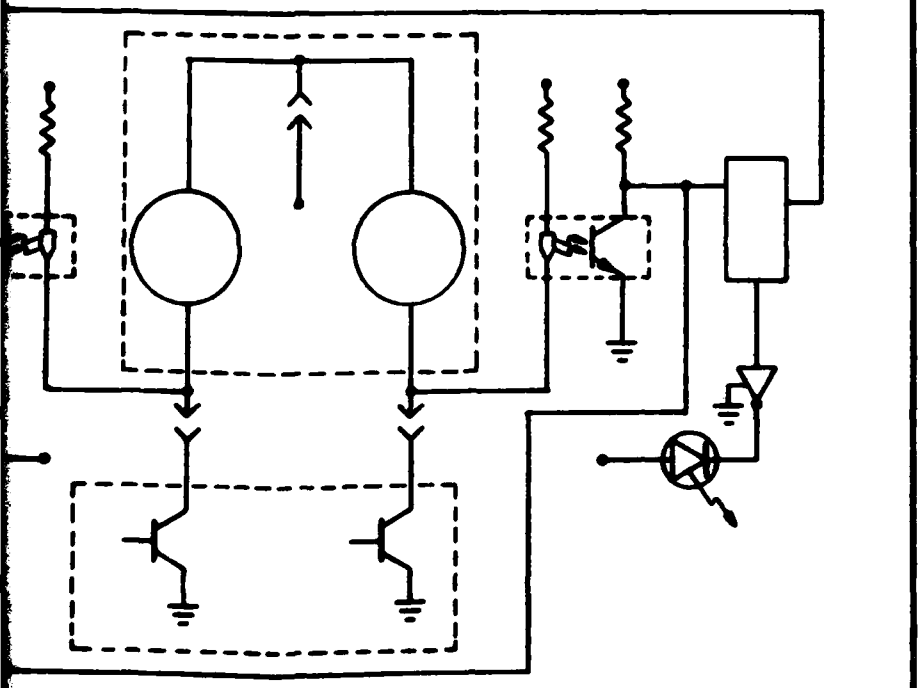


V8A V8B
STD



74279

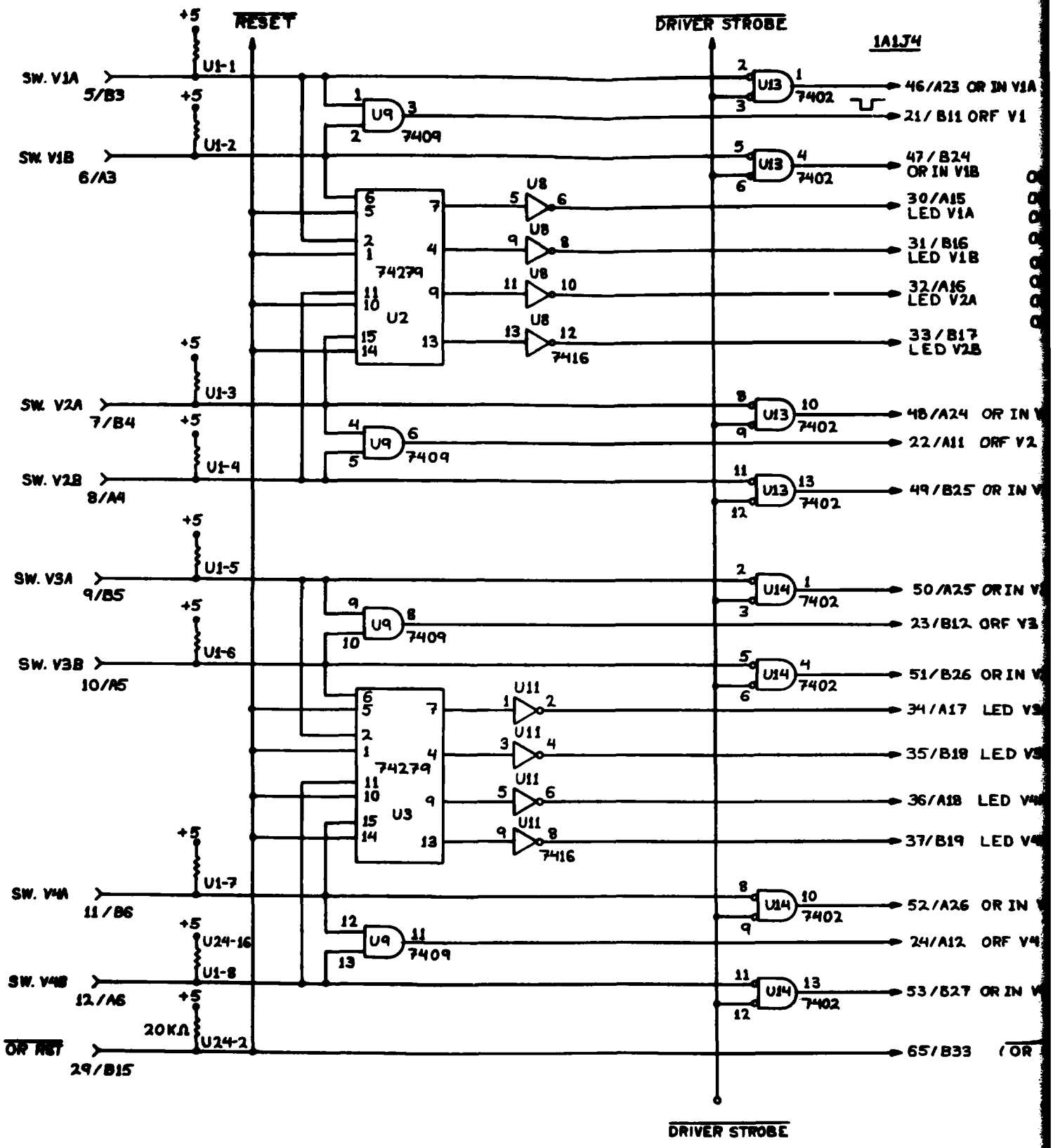




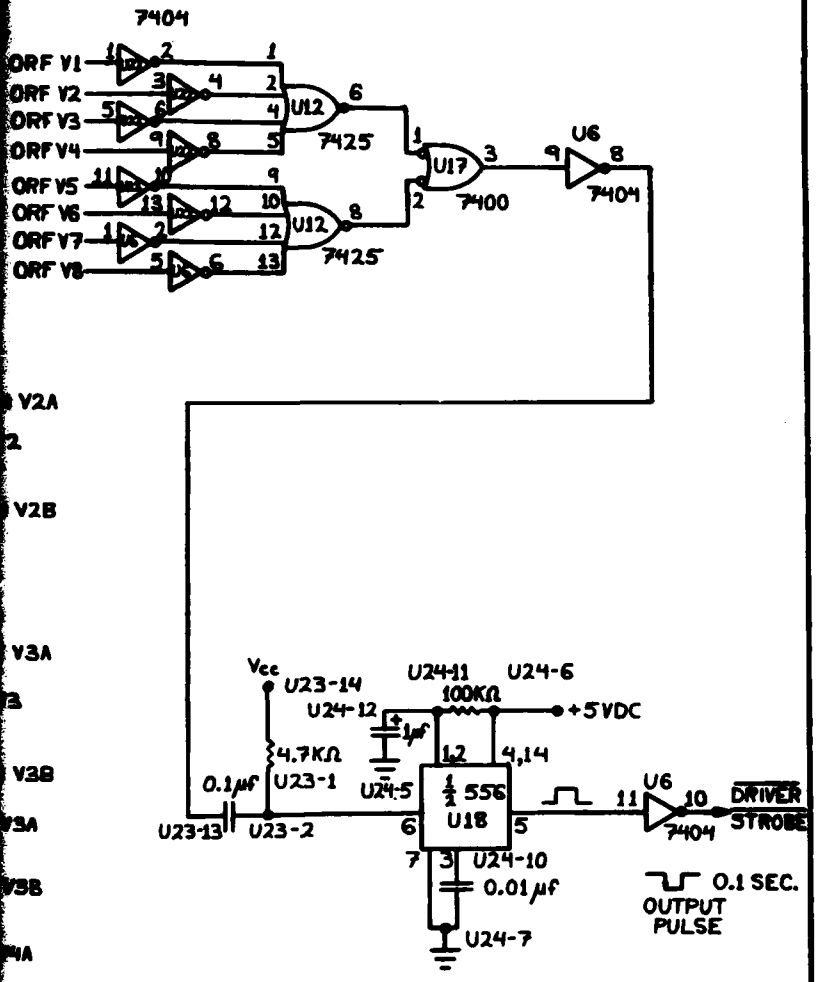
DESIGN C. CLEMENTS	NOSC	DRAFT JMF
DRW# CC-28	9-7-79	REV# 1
LED DISPLAY OF VALVE POSITION		SHT 2 OF 2

1

2


RESET
DRIVER STROBE
1A1J4
+5
U1-1
SW. V1A
5/B3
U1-2
SW. V1B
6/A3
U1-3
SW. V2A
7/B4
U1-4
SW. V2B
8/A4
U1-5
SW. V3A
9/B5
U1-6
SW. V3B
10/A5
U1-7
SW. V4A
11/B6
U24-16
U1-8
SW. V4B
12/A6
20KΩ
U24-2
OR RST
29/B15
U9
7409
U2
74279
U3
U8
U11
7416
U13
U14
7402
46/A23 OR IN V1A
21/B11 ORF V1
47/B24 OR IN V1B
30/A15 LED V1A
31/B16 LED V1B
32/A16 LED V2A
33/B17 LED V2B
48/A24 OR IN V2
22/A11 ORF V2
49/B25 OR IN V2
50/A25 OR IN V3
23/B12 ORF V3
51/B26 OR IN V3
34/A17 LED V3
35/B18 LED V3
36/A18 LED V4
37/B19 LED V4
52/A26 OR IN V4
24/A12 ORF V4
53/B27 OR IN V4
65/B33 (OR
DRIVER STROBE

MOD. 5-9-79



V2A

2

V2B

V3A

3

V3B

V3A

V3B

V4A

V4B

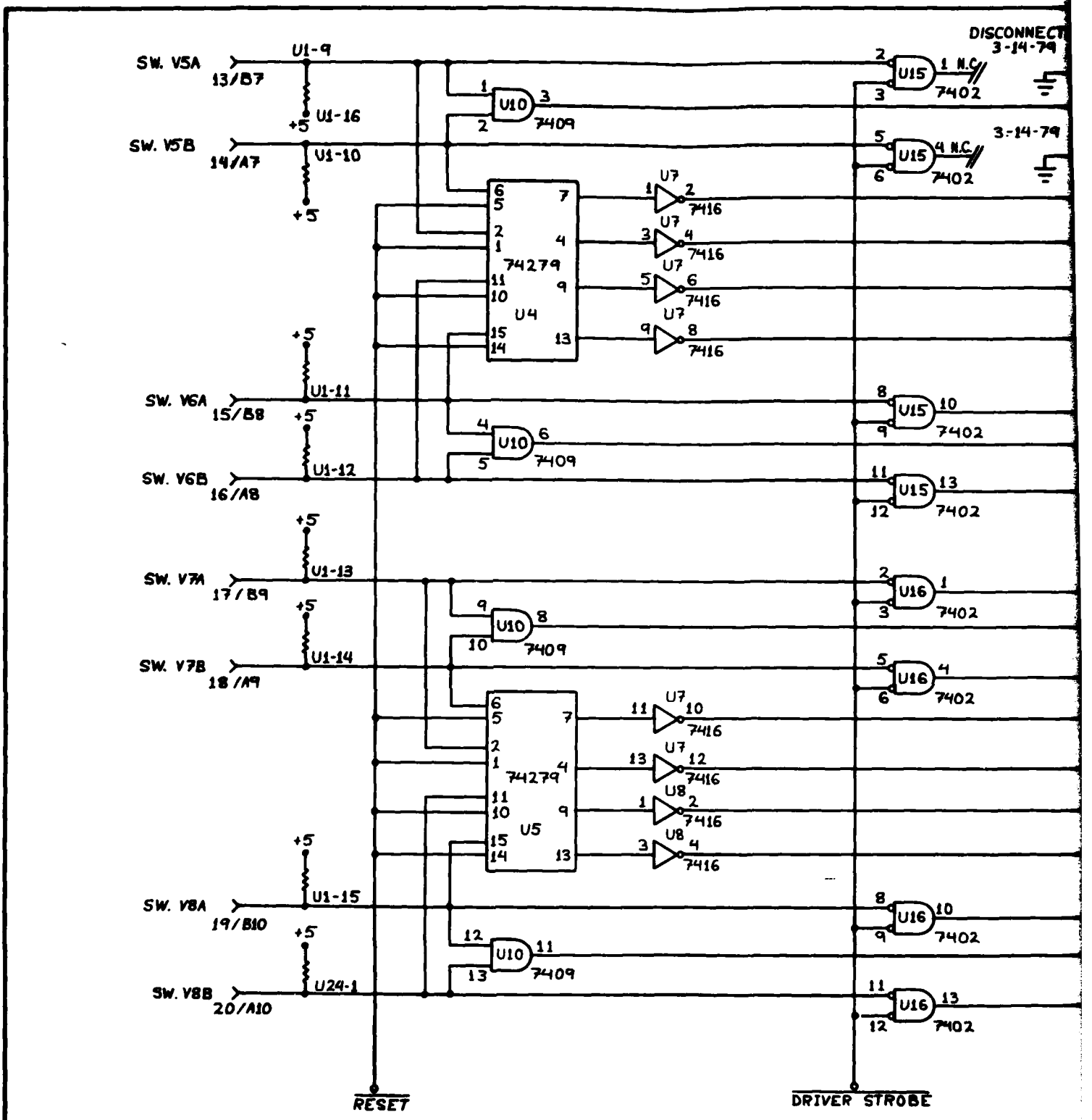
V4A OUTPUT TO VALVE

V4 (OVER-RIDE FLAG OUTPUT)

V4B OUTPUT TO VALVE

RST)

DESIGN C. CLAWSON	NOSC	DRAFT JMF.
DRAWING NO. CC-3A	5-9-79	REV. 2
MANUAL VALVE OVERRIDE INTERFACE BOARD		
		SHEET 1 OF 2


DISCONNECT
3-14-79
SW. V5A
13/B7
U1-9
+5
U1-16
SW. V5B
14/A7
U1-10
+5
U10
7409
U4
74279
U7
7416
U15
7402
1 N.C.
4 N.C.
3-14-79
SW. V6A
15/B8
U1-11
SW. V6B
16/A8
U1-12
SW. V7A
17/B9
U1-13
SW. V7B
18/A9
U1-14
U5
U8
SW. V8A
19/B10
U1-15
SW. V8B
20/A10
U24-1
U16
RESET
DRIVER STROBE

ECTED

79

54/A27 OR IN V5A

25/B13 ORF V5

79

55/B28 OR IN V5B

38/A19 LED V5A

39/B20 LED V5B

40/A20 LED V6A

41/B21 LED V6B

56/A28 OR IN V6A

26/A13 ORF V6

57/B29 OR IN V6B

58/A29 OR IN V7A

27/B14 ORF V7

59/B30 OR IN V7B

42/A21 LED V7A

43/B22 LED V7B

44/A22 LED V8A

45/B23 LED V8B

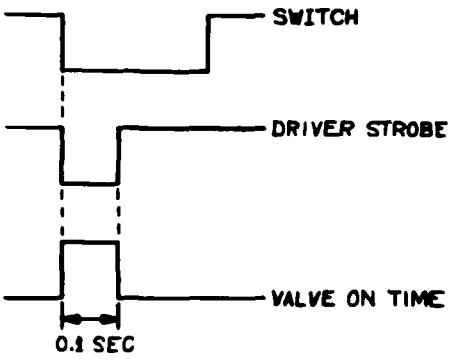
60/A30 OR IN V8A

28/A14 ORF V8

61/B31 OR IN V8B

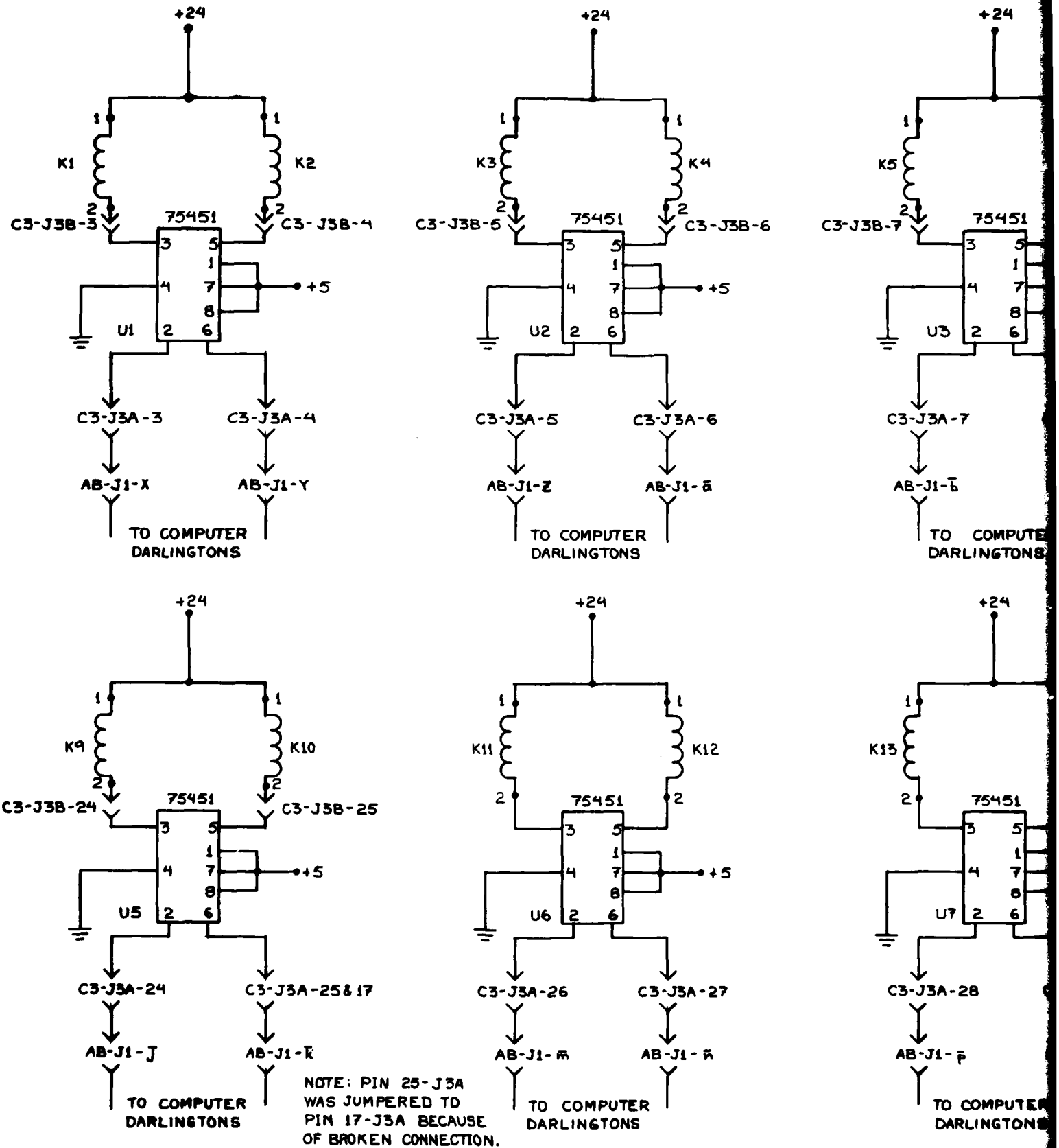
KEY

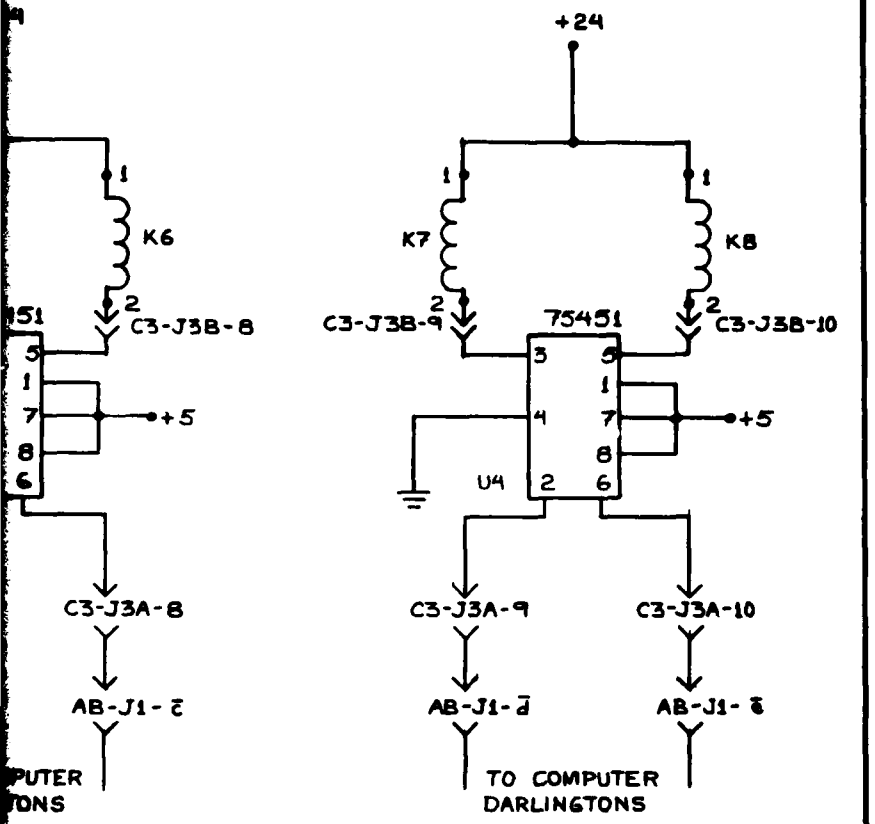
BOARD PIN # / EDGE CONN. #

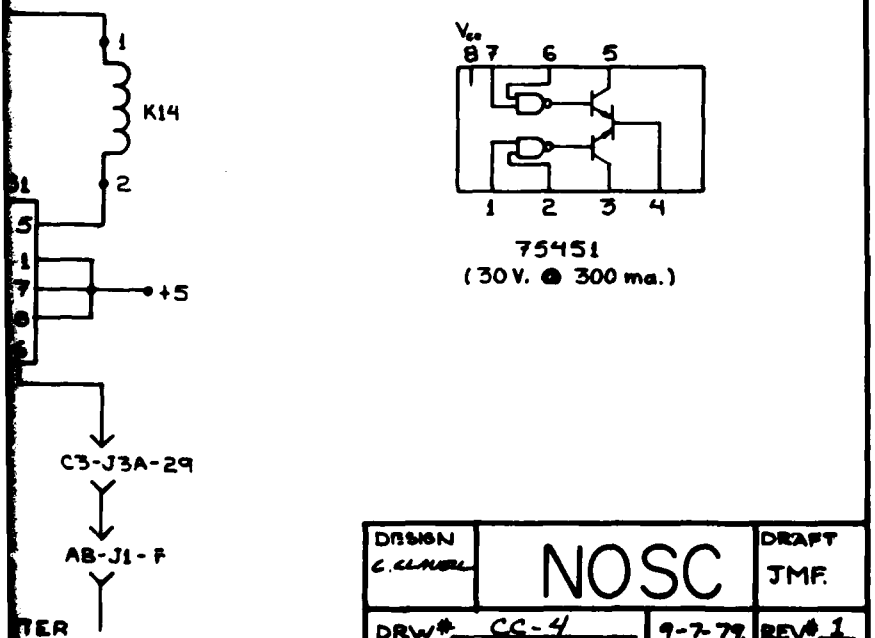


DESIGN C. CLAVELL	NOSC	DRAFT JMF
DRAWING NO. CC-38	5-5-78	REV. 2
MANUAL VALVE OVERRIDE INTERFACE BOARD		SHEET 2 OF 2

2

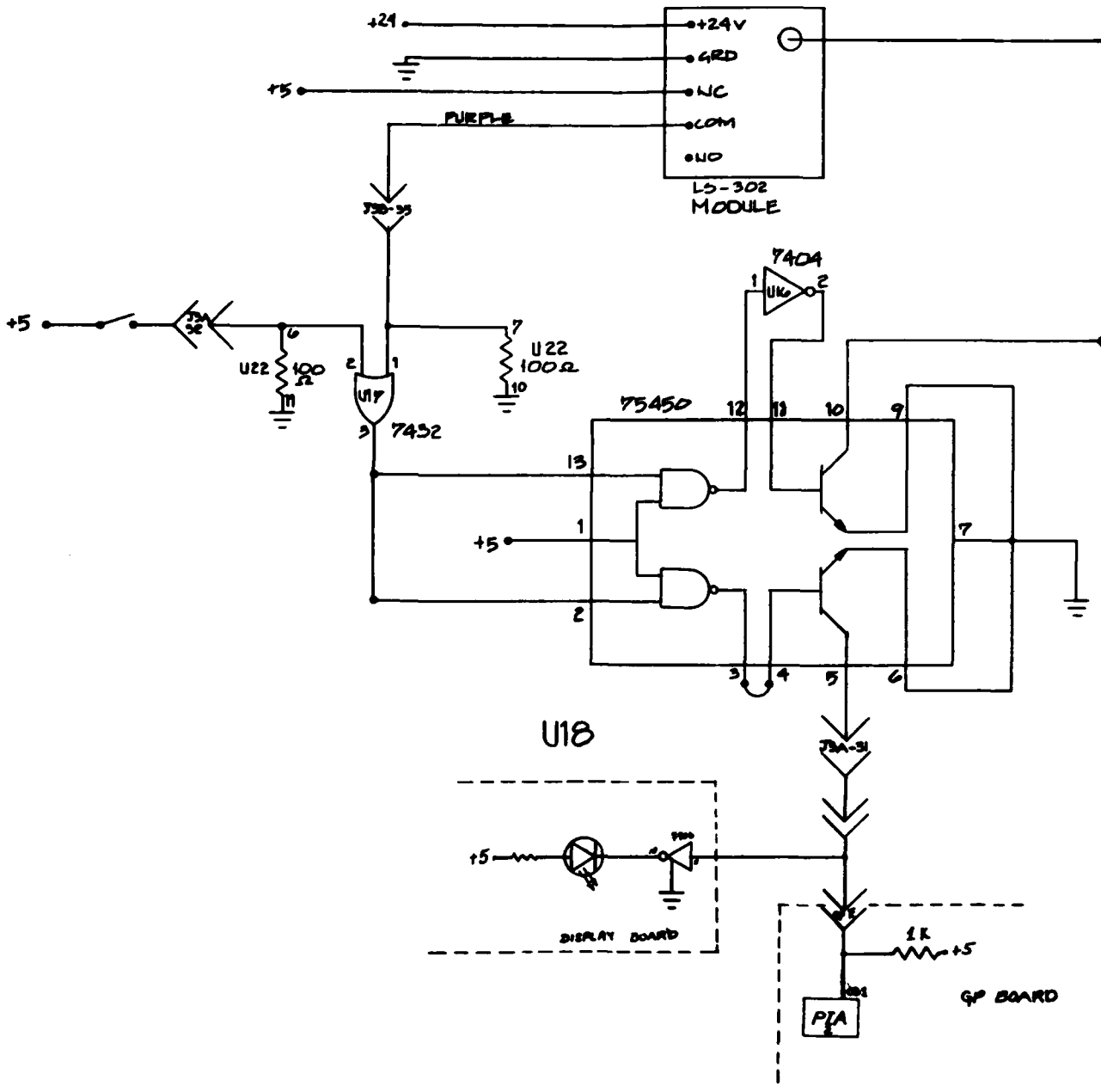

+24
K1
K2
C3-J3B-3
C3-J3B-4
75451
U1
+5
C3-J3A-3
C3-J3A-4
AB-J1-X
AB-J1-Y
TO COMPUTER DARLINGTONS
K3
K4
C3-J3B-5
C3-J3B-6
U2
C3-J3A-5
C3-J3A-6
AB-J1-Z
AB-J1-a
K5
C3-J3B-7
U3
C3-J3A-7
AB-J1-b
K9
K10
C3-J3B-24
C3-J3B-25
U5
C3-J3A-24
C3-J3A-25&17
AB-J1-J
AB-J1-k
NOTE: PIN 25-J3A WAS JUMPERED TO PIN 17-J3A BECAUSE OF BROKEN CONNECTION.
K11
K12
C3-J3B-26
C3-J3B-27
U6
C3-J3A-26
C3-J3A-27
AB-J1-m
AB-J1-n
K13
U7
C3-J3A-28
AB-J1-p





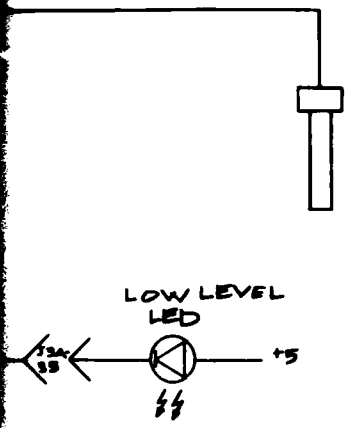
DESIGN C. CLARK	NOSC	DRAFT JMF.
DRW# CC-4	9-7-79	REV# 1
COMPUTER TO RELAY INTERFACE		
	SHT 1 OF 1	

ON LD-L3
& LS-302 MODULE
(LORION SCIENTIFC)

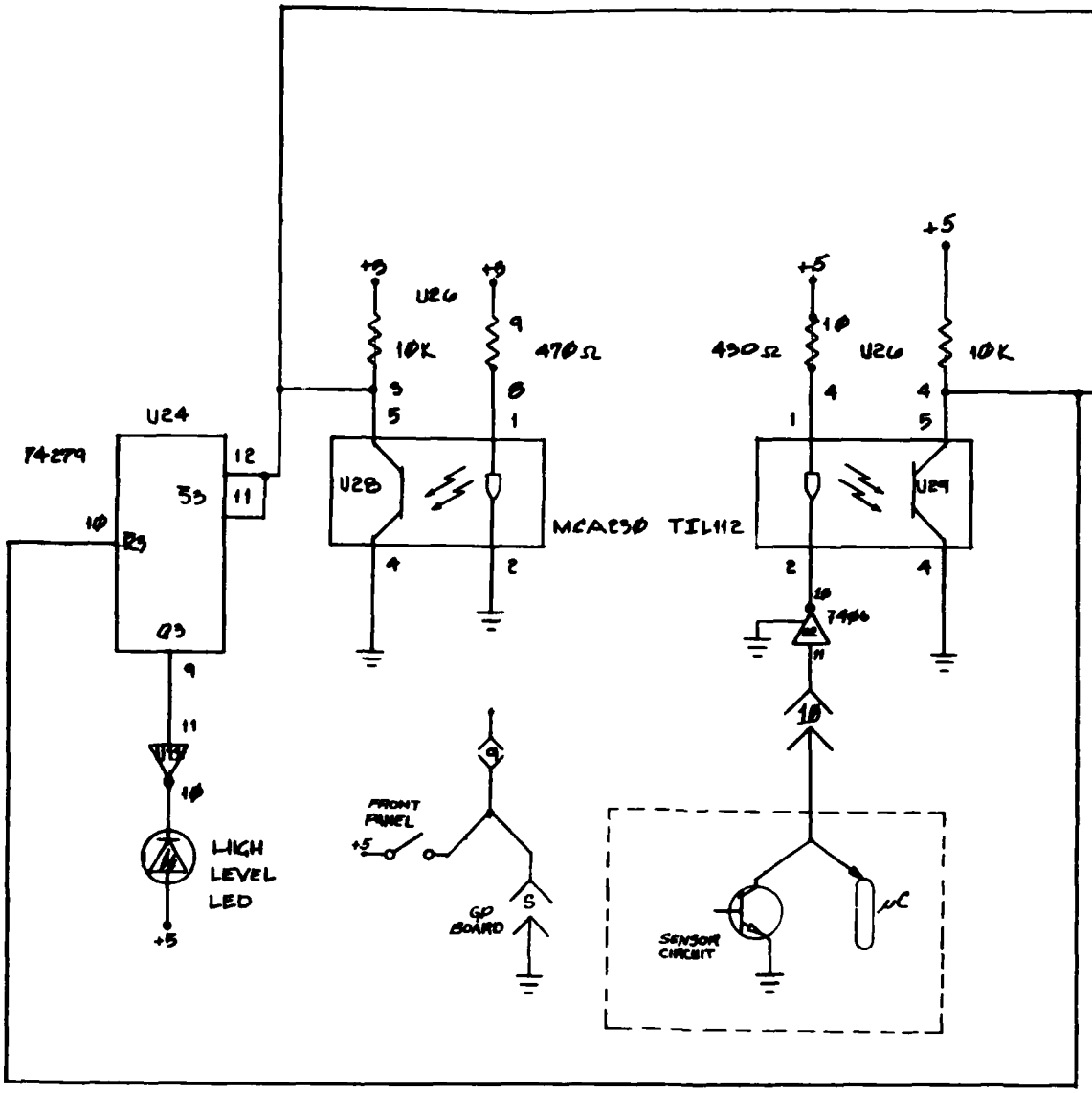


ACT. CONDITIONS

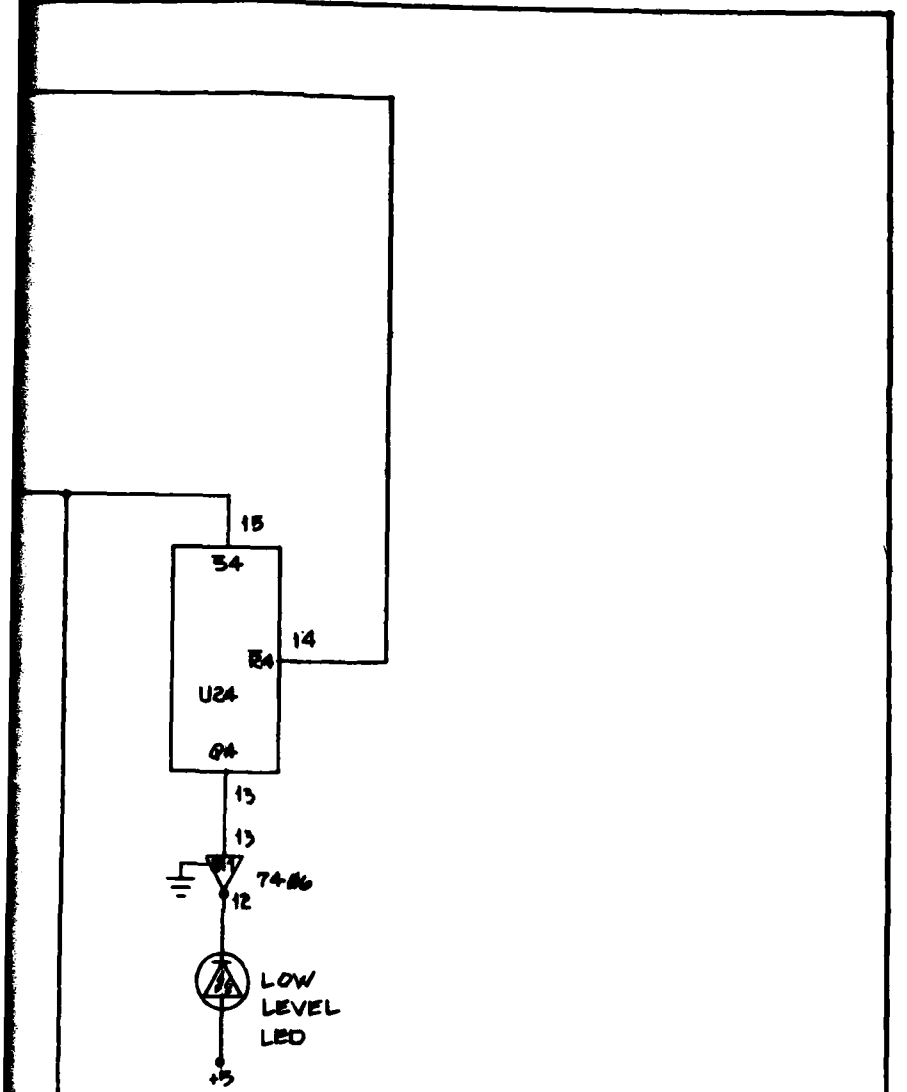
NC: OPEN - WET
NC: CLOSE - DRY



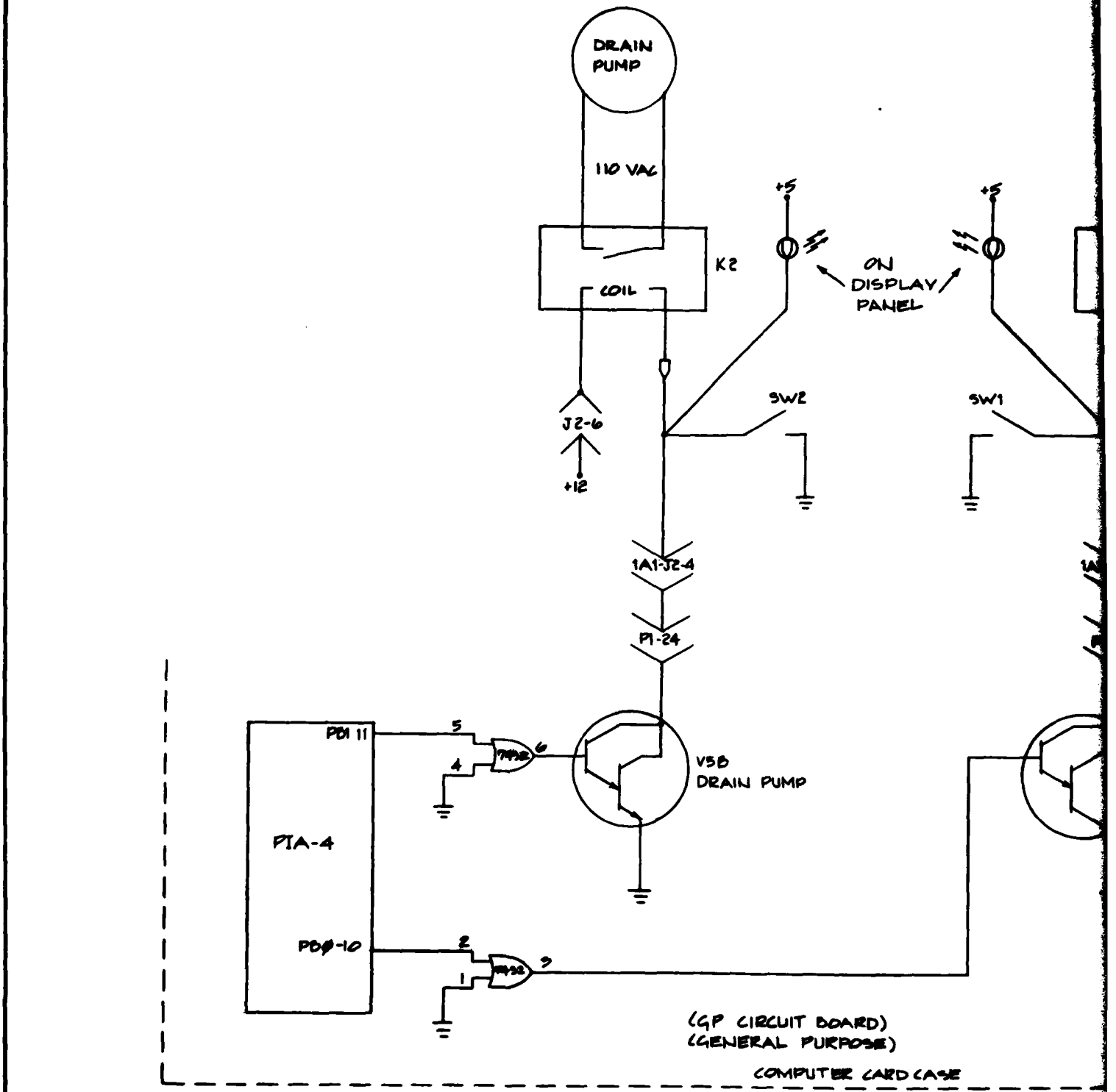
DESIGN C. CLAVELL	NOSC	DRAFT EMK
DRW# CC-5	7-9-79	REV.# 1
LOW LEVEL SENSOR CIRCUIT		SHT 1 OF 1

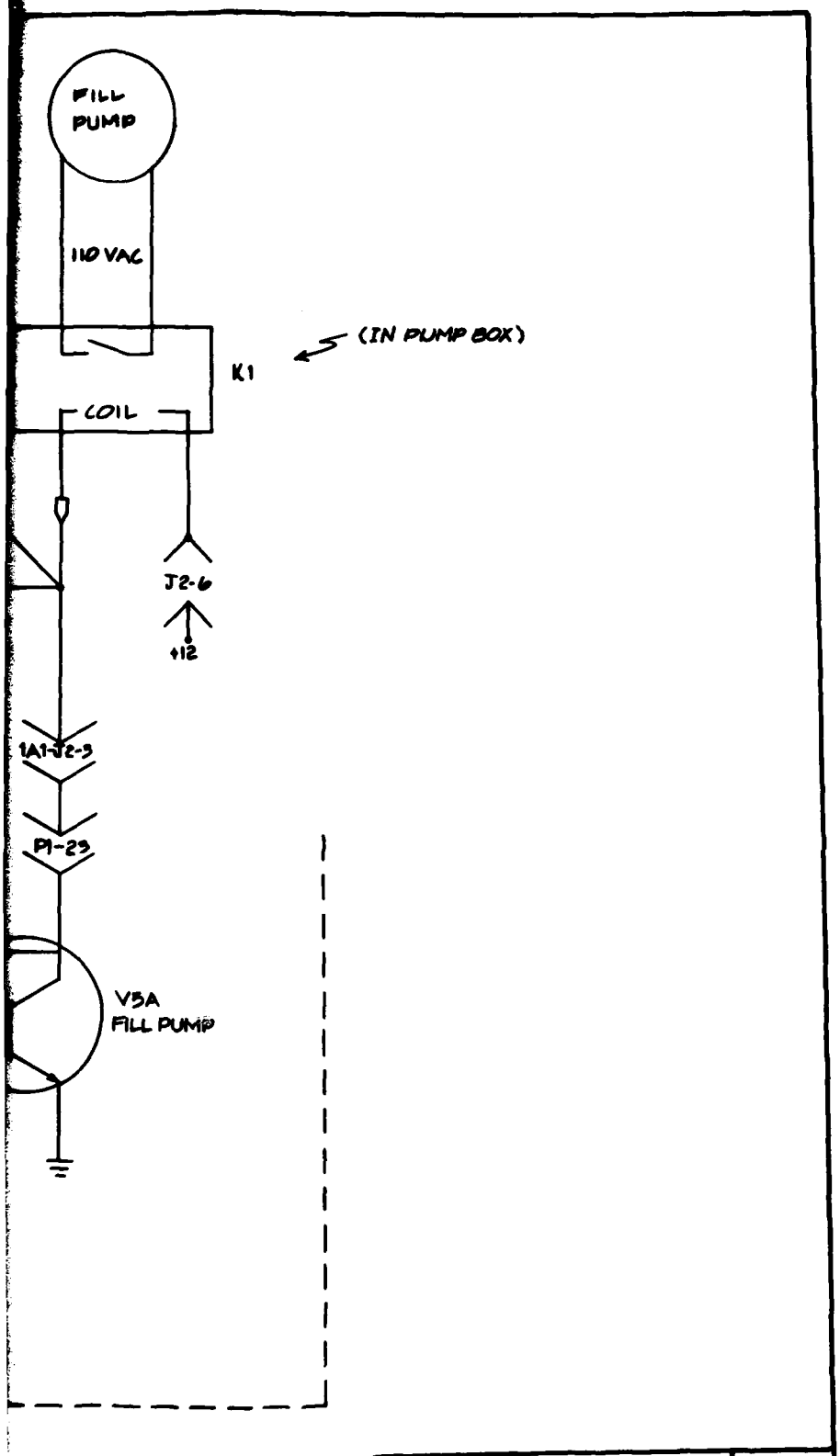


[PART OF
PANEL DISPLAY BOARD]

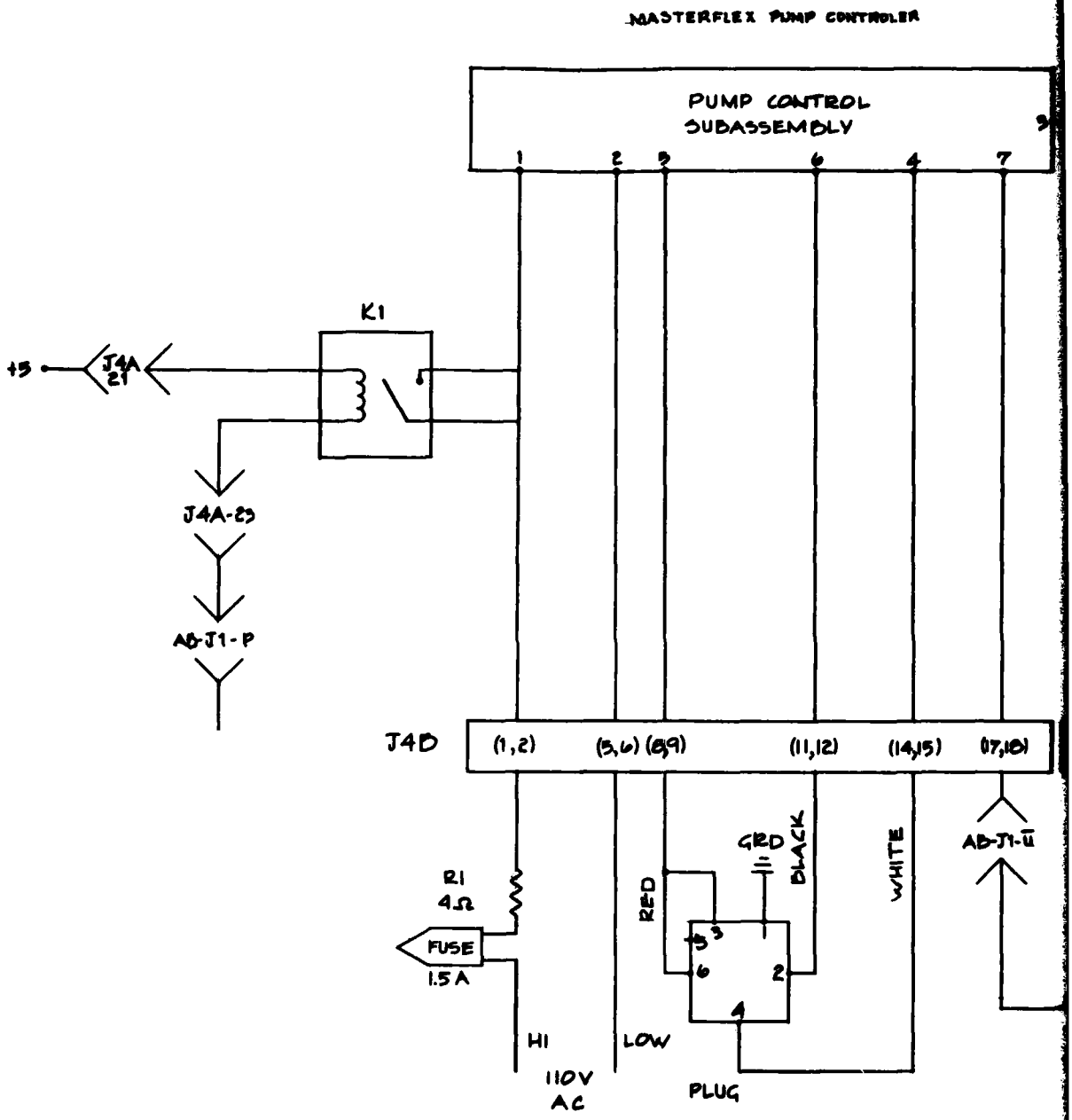


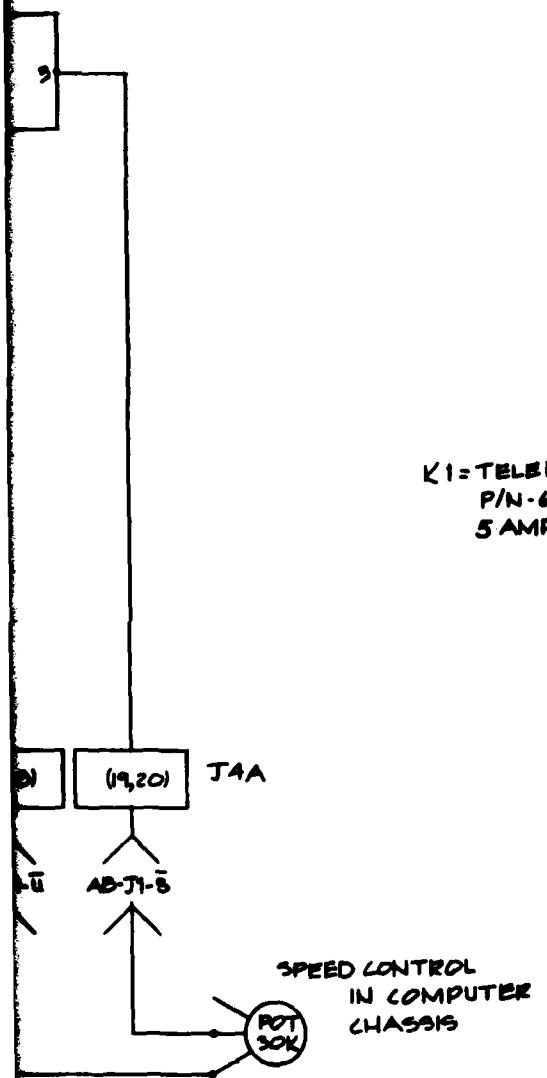
DESIGN C. CLEMENT	NOSC	DRAFT EMK
DRW. # CC-6	9-7-79	REV.# 1
HIGH & LOW WATER LEVEL SENSOR DISPLAY CIRCUIT		SHT 1 OF 1


DRAIN PUMP
110 VAC
K2
COIL
+5
+5
ON DISPLAY PANEL
SW2
SW1
J2-6
+12
1A1-J2-4
P1-24
P1 11
5
4
6
V5B
DRAIN PUMP
PIA-4
P0Ø-10
2
1
3
(GP CIRCUIT BOARD)
(GENERAL PURPOSE)
COMPUTER CARD CASE



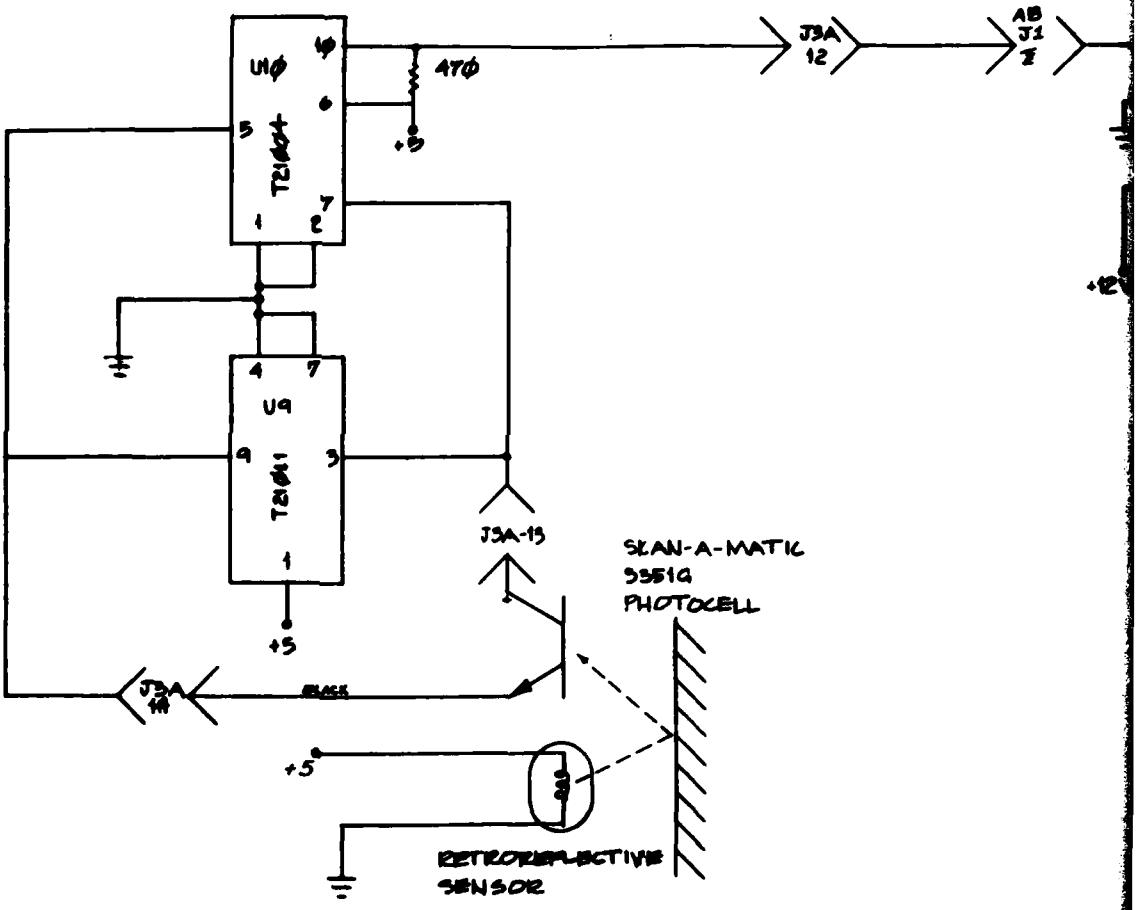
DESIGN C. CLAWSON	NOSC	DRAFT EMK
DRW# CC-7	9-7-79	REV.# 1
FILL & EMPTY PUMP CIRCUIT		SHT 1 OF 1


MASTERFLEX PUMP CONTROLER
PUMP CONTROL
SUBASSEMBLY
1
2
3
6
4
7
K1
+5
J4A
21
J4A-23
AB-J1-P
J4D
(1,2)
(5,6)
(8,9)
(11,12)
(14,15)
(17,18)
AB-J1-11
R1
4Ω
FUSE
1.5A
RED
GRD
BLACK
WHITE
5
6
2
4
HI
LOW
110V
AC
PLUG

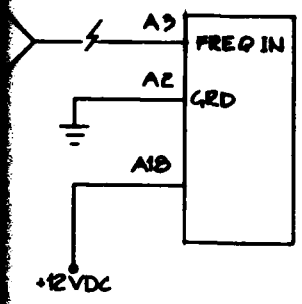


K1 = TELEDYNE
P/N-601-1401
5 AMP-AC

DESIGN C. CLARKE	NOSC	DRAFT EMK.
DRW# CC-8	9-7-72	REV# 1
PUMP CONTROLLER C4		
	SHT 1 OF 1	


U10
TRIGGER
470
+5
J3A
12
AB
J1
2
+12
U9
TRIGGER
J3A-13
SLAN-A-MATIC
5551Q
PHOTOCELL
J3A
14
+5
RETROREFLECTIVE
SENSOR

FREQ METER (6 DIGIT)
IMC MODEL: 4001030

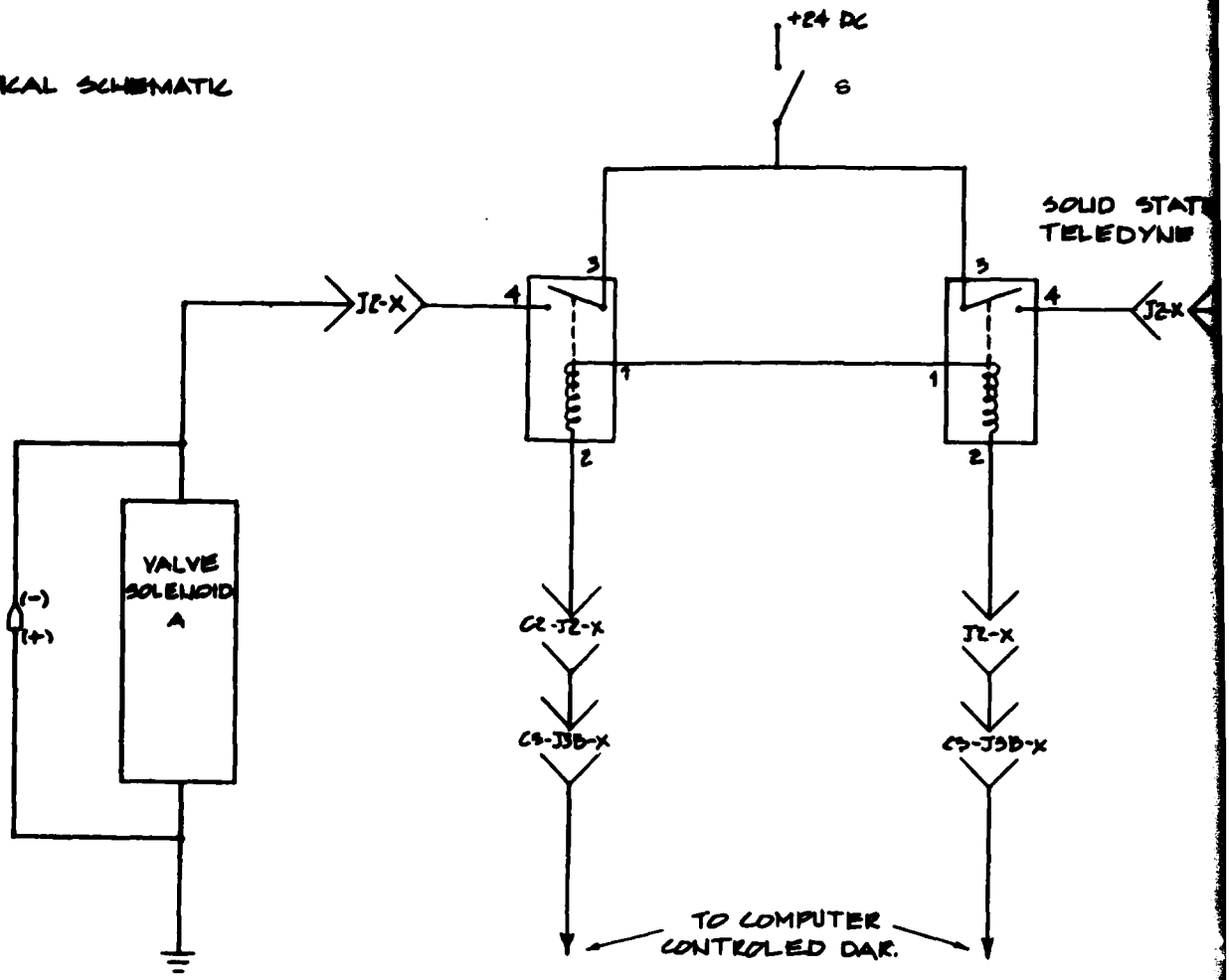


FREQ. METER
READS IN RPS

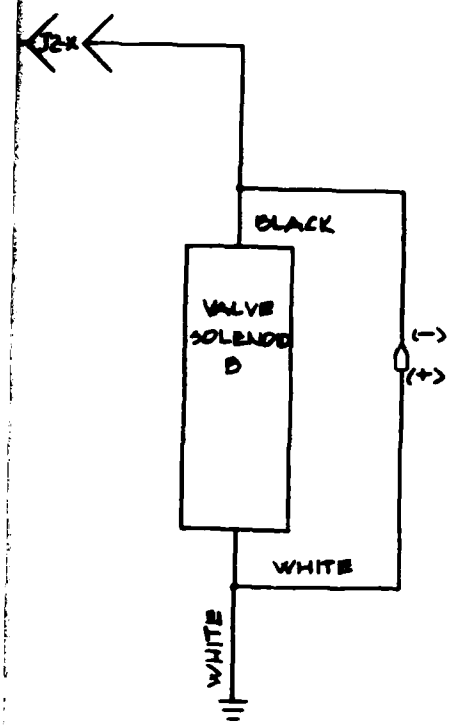
SKAN-A-MATIC CORP
MODULES & PHOTOCELL

DESIGN C. CLAVELL	NOSC	DRAFT EMK.
DRW# CC-9	9-7-79	REV# 1
RPS SENSOR CIRCUIT CARD 3		SHT 1 OF 3

TYPICAL SCHEMATIC

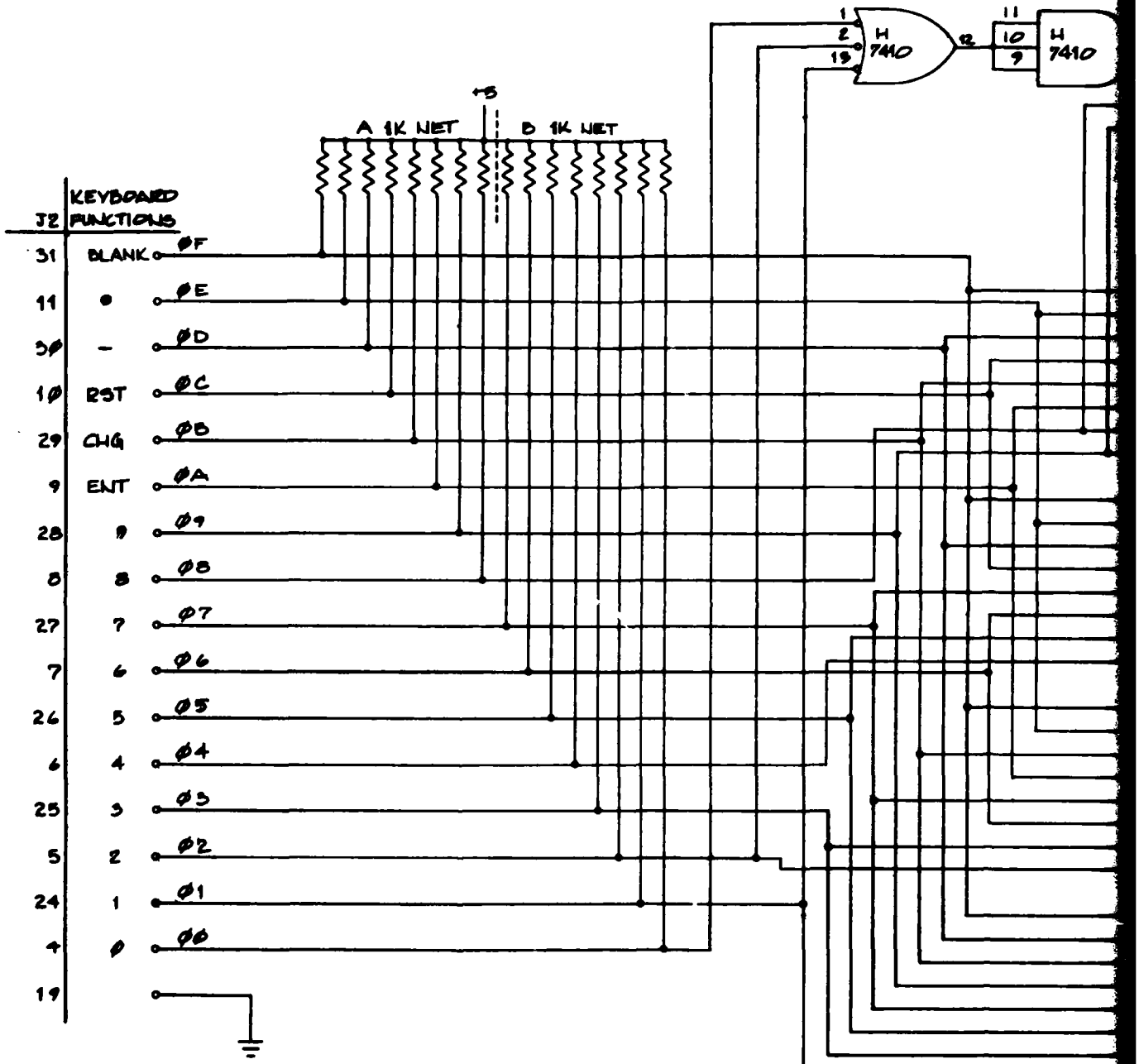


STATE RELAYS - 5AMP DC
DYNE 603-2P

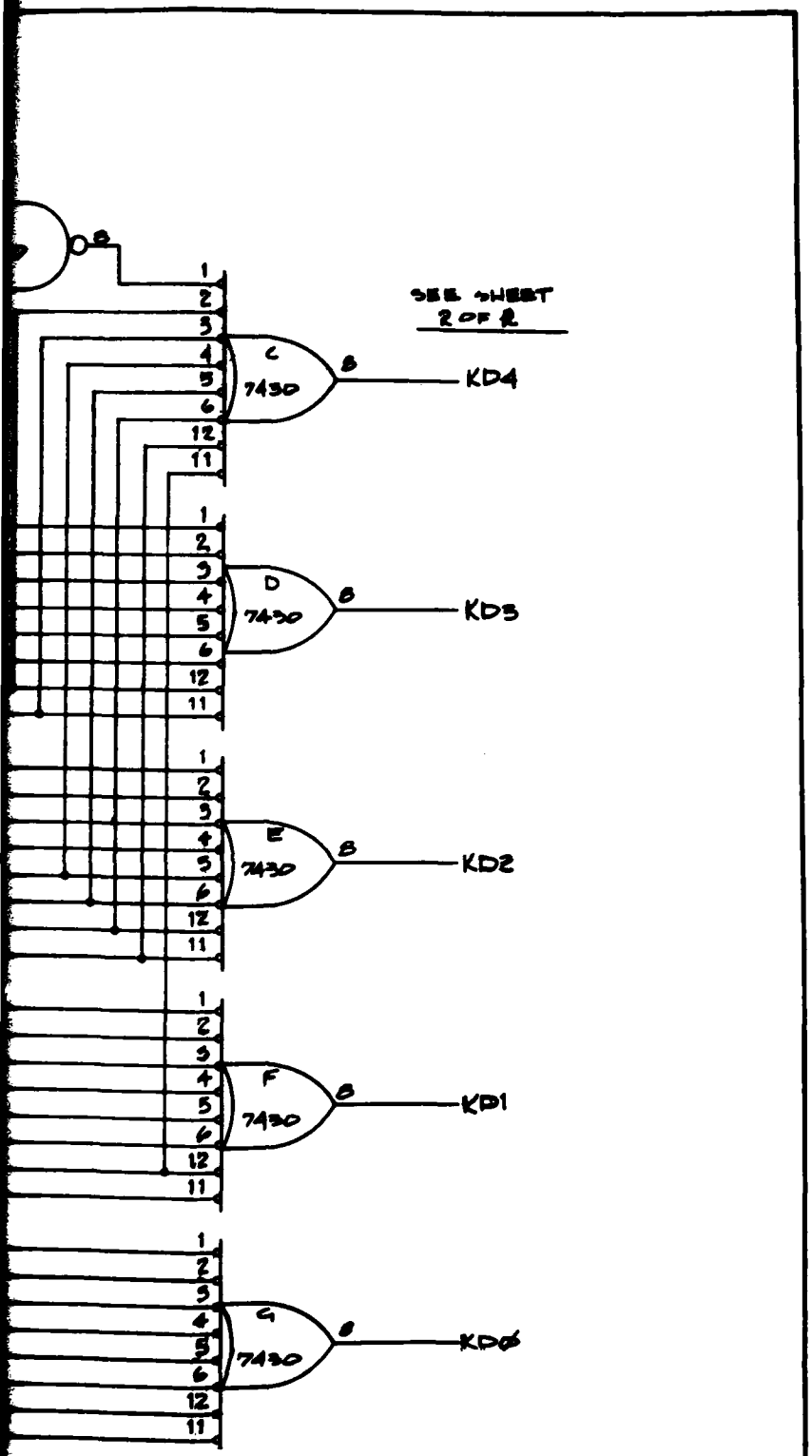


SOLENOIDS: DELTROL
24 VDC - PULSE
#53623-83

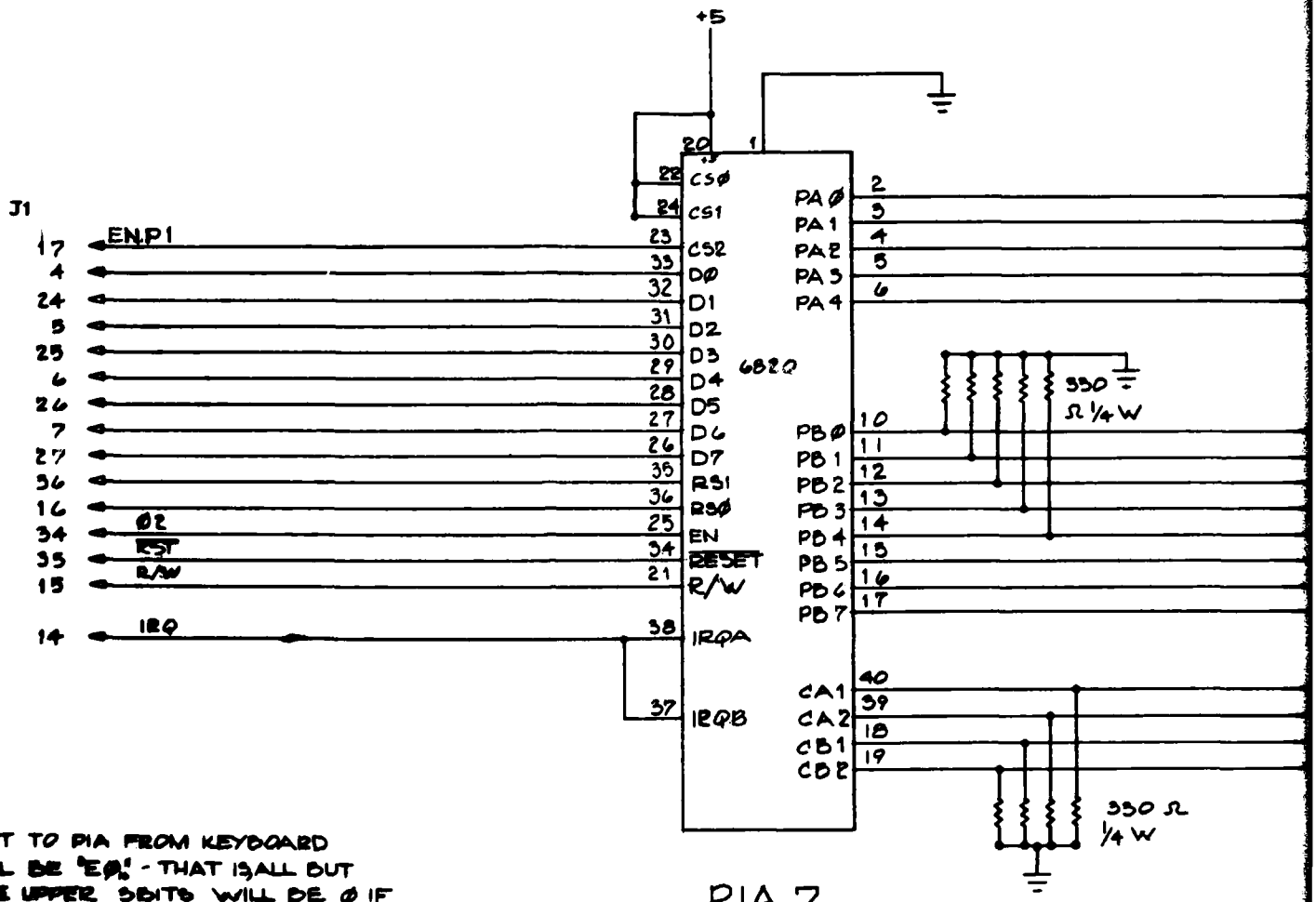
DESIGN CLEANER	NOSC	DRAFT EMK
DRW # CC-10	9-7-79	REV# 1
MAIN VALVE BOARD RELAY BOARD CARD 2	SHT 1 OF 1	



J2
GND 19,20
39,40



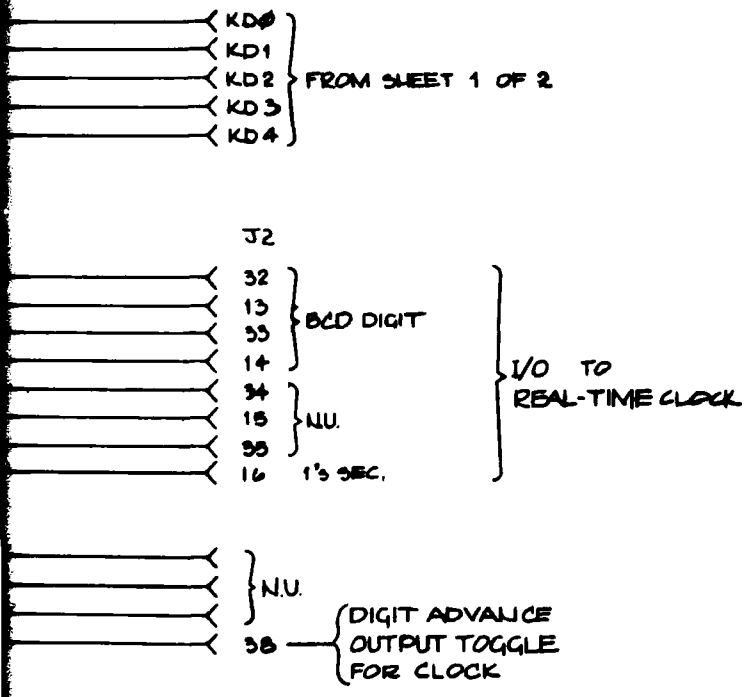
DESIGN C. CLAVELL	NOSC	DRAFT EMK
DRW# CC-11A	9-7-79	REV# 1
SEA WATER ANALYZER EXTRA BOARD 1	SHT 1 OF 2	



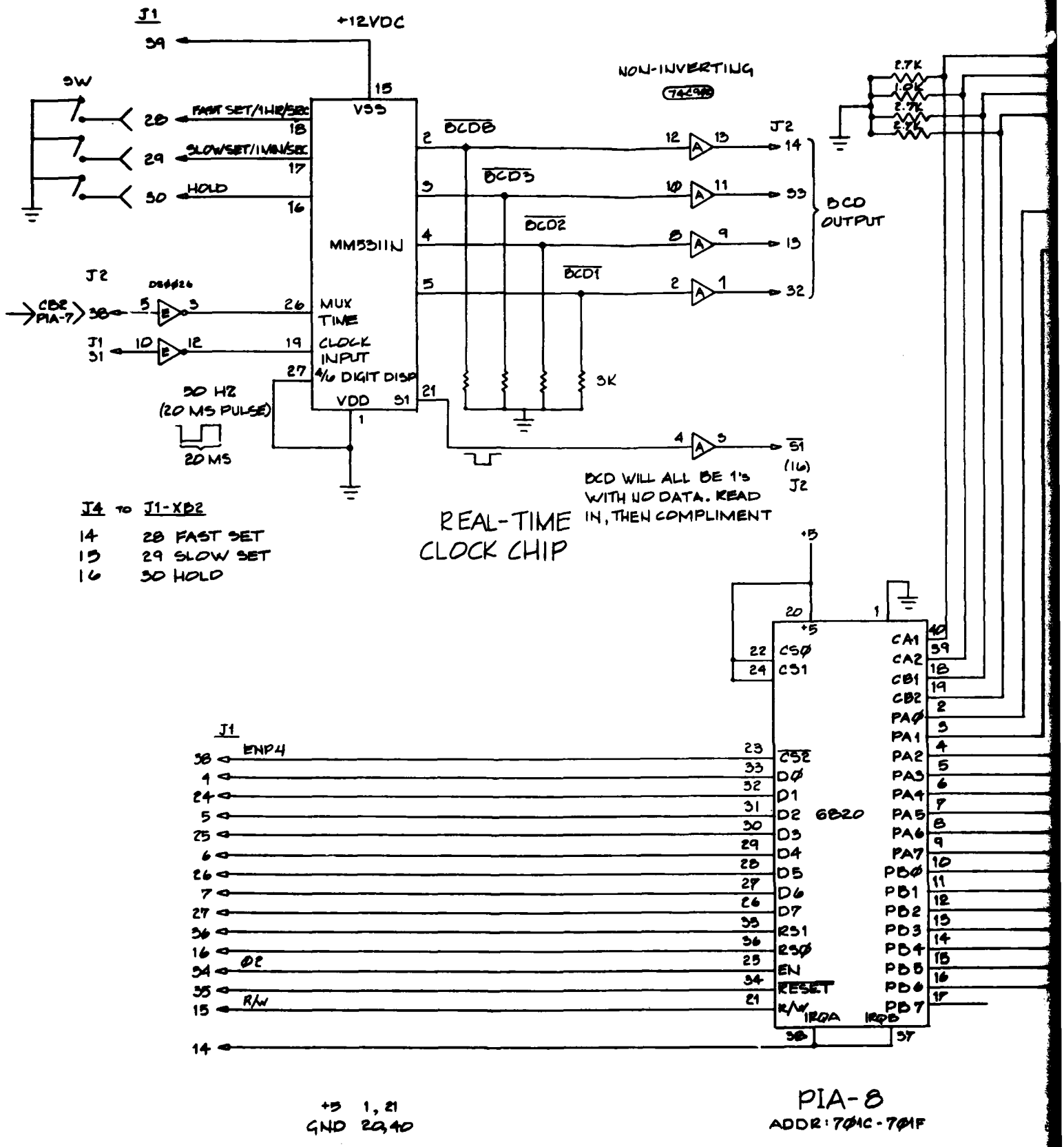
INPUT TO PIA FROM KEYBOARD WILL BE 'EØ' - THAT IS, ALL BUT THE UPPER 3 BITS WILL BE Ø IF NO KEY IS HELD DOWN

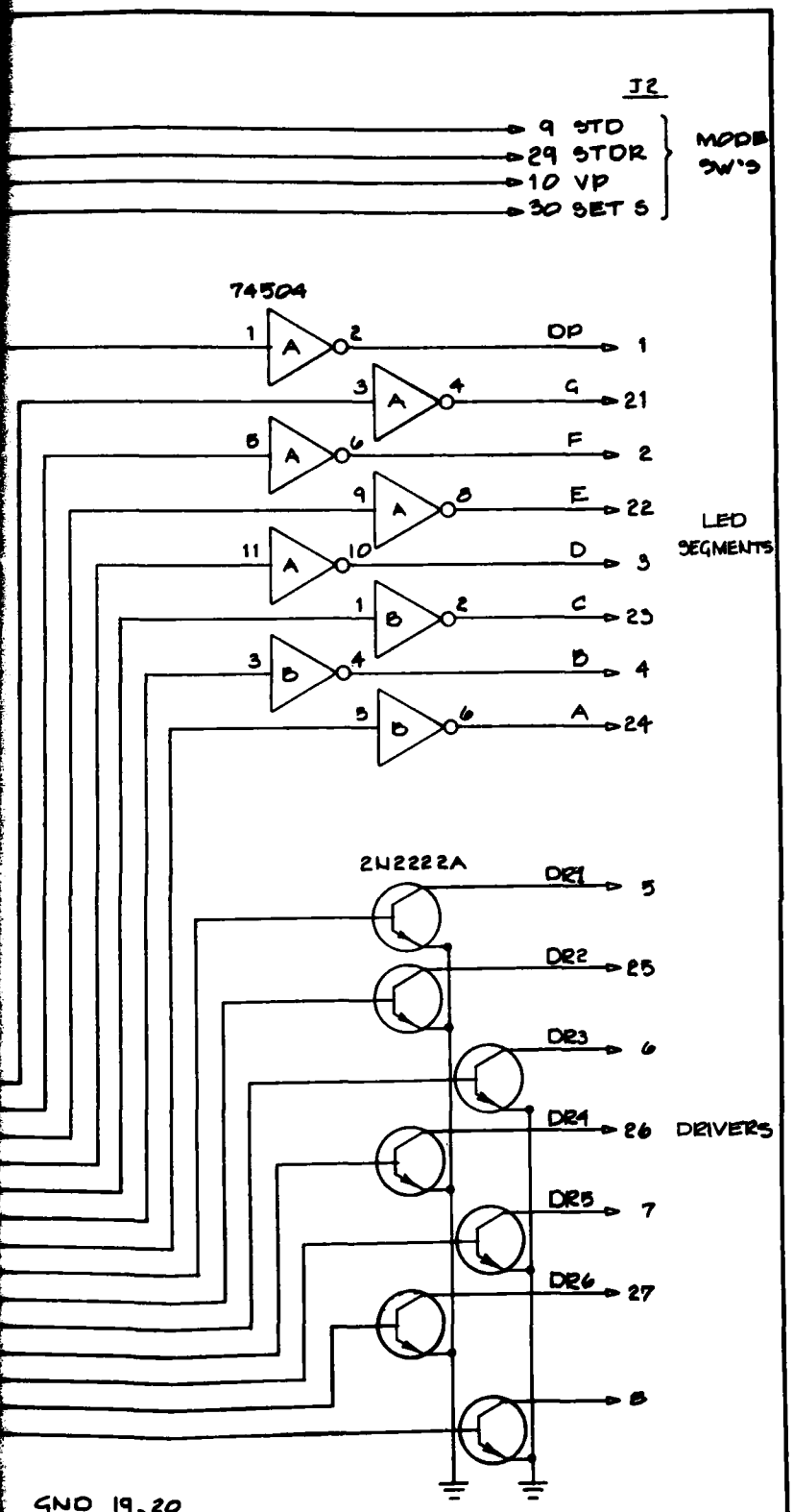
PIA 7
ADDR: 7ØIØ-7ØI3

J1
+5 1, 21
GND 20, 40



DESIGN CLANELL	NOSC	DRAFT EMK
DRW# CC-11B	9-7-79	REV# 1
SEA WATER ANALYZER EXTRA BOARD 1		SHT 2 OF 2

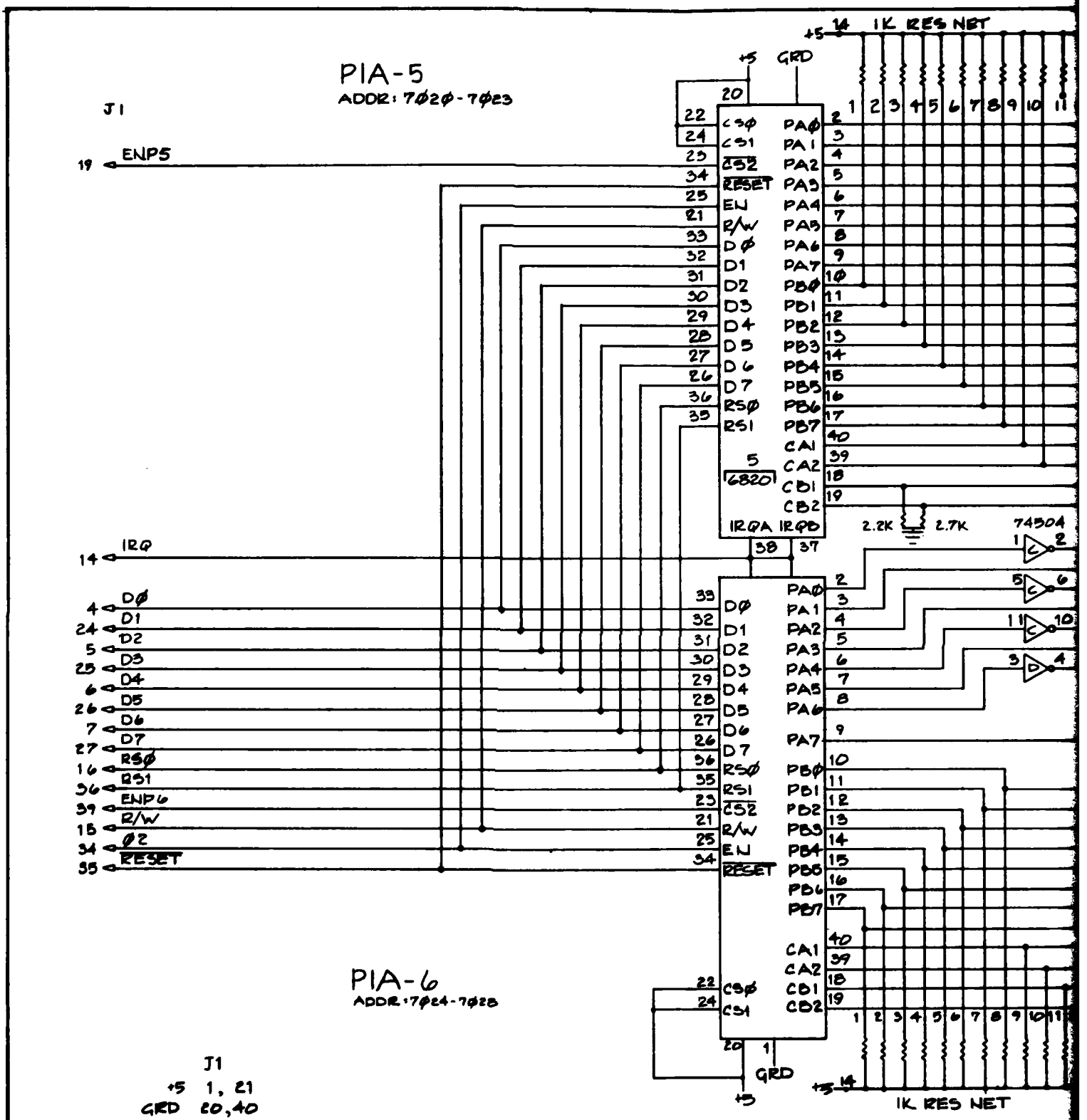

J1
+12VDC
39
SW
15
VSS
28
FAST SET/1HR/SEC
18
29
SLOW SET/1MIN/SEC
17
30
HOLD
16
MM5311N
NON-INVERTING
74LS98
2.7K
BCD8
BCD4
BCD2
BCD1
J2
14
33
15
32
BCD OUTPUT
3K
J2
DS8926
CBR PIA-7
38
26
MUX TIME
J1
31
19
CLOCK INPUT
27
4½ DIGIT DISP
50 HZ
(20 MS PULSE)
20 MS
VDD
21
1
S1
(16)
J2
BCD WILL ALL BE 1's WITH NO DATA. READ IN, THEN COMPLIMENT
J4 TO J1-XB2
14 28 FAST SET
15 29 SLOW SET
16 30 HOLD
REAL-TIME CLOCK CHIP
+5
20
1
22
24
CS0
CS1
CA1
CA2
CB1
CB2
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
J1
38
ENP4
4
24
5
25
6
26
7
27
36
16
34
35
15
14
Ø2
R/W
23
33
32
31
30
29
28
27
26
35
36
25
34
21
CS2
D0
D1
D2
D3
D4
D5
D6
D7
RS1
RS0
EN
RESET
R/W
6820
IRQA
IRQB
38
37
40
39
18
19
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
+5 1, 21
GND 20,40
PIA-8
ADDR: 701C - 701F

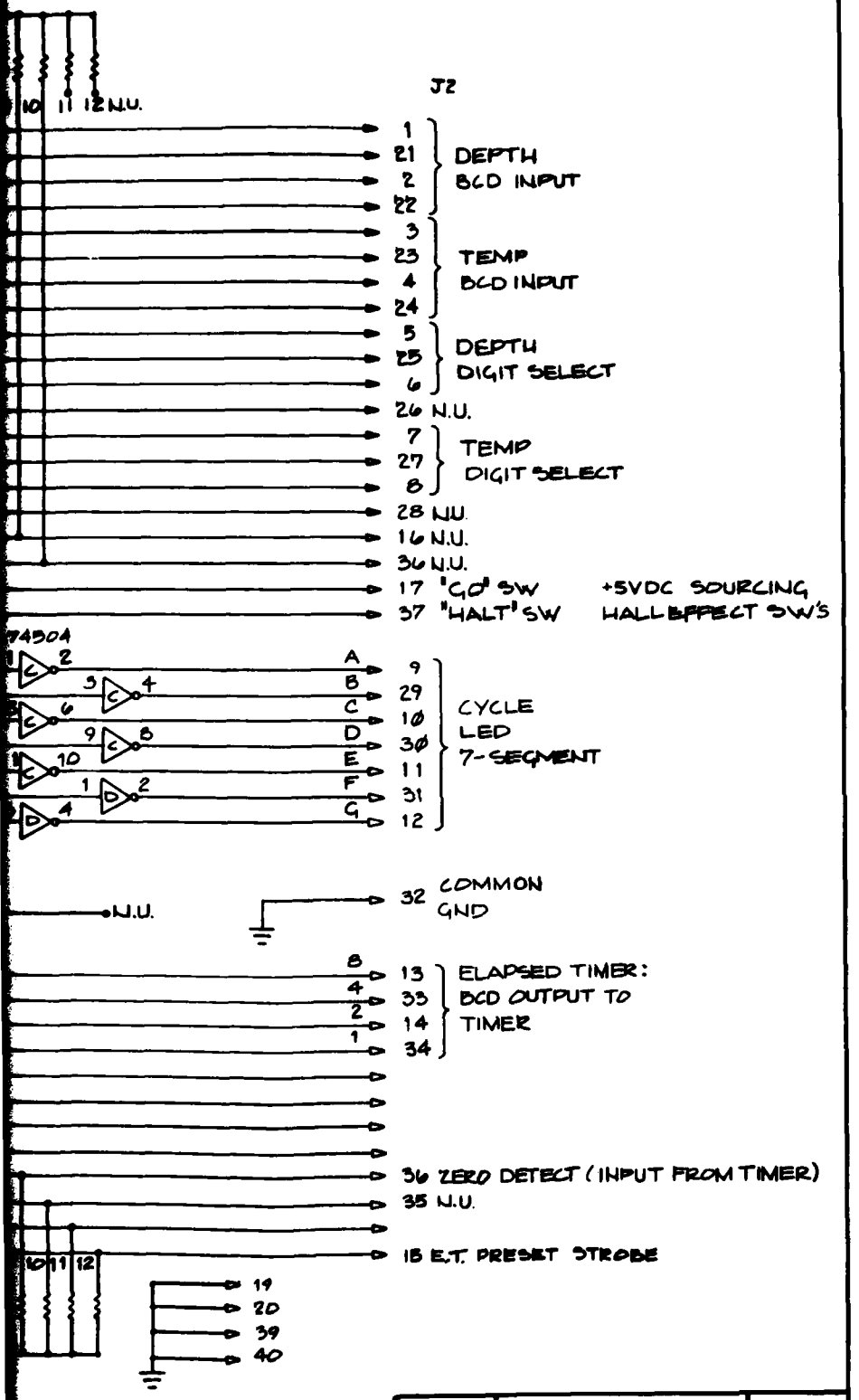


DESIGN C. CLARK	NOSC	DRAFT EMK
DRW# CC-12	9-7-79	REV# 1

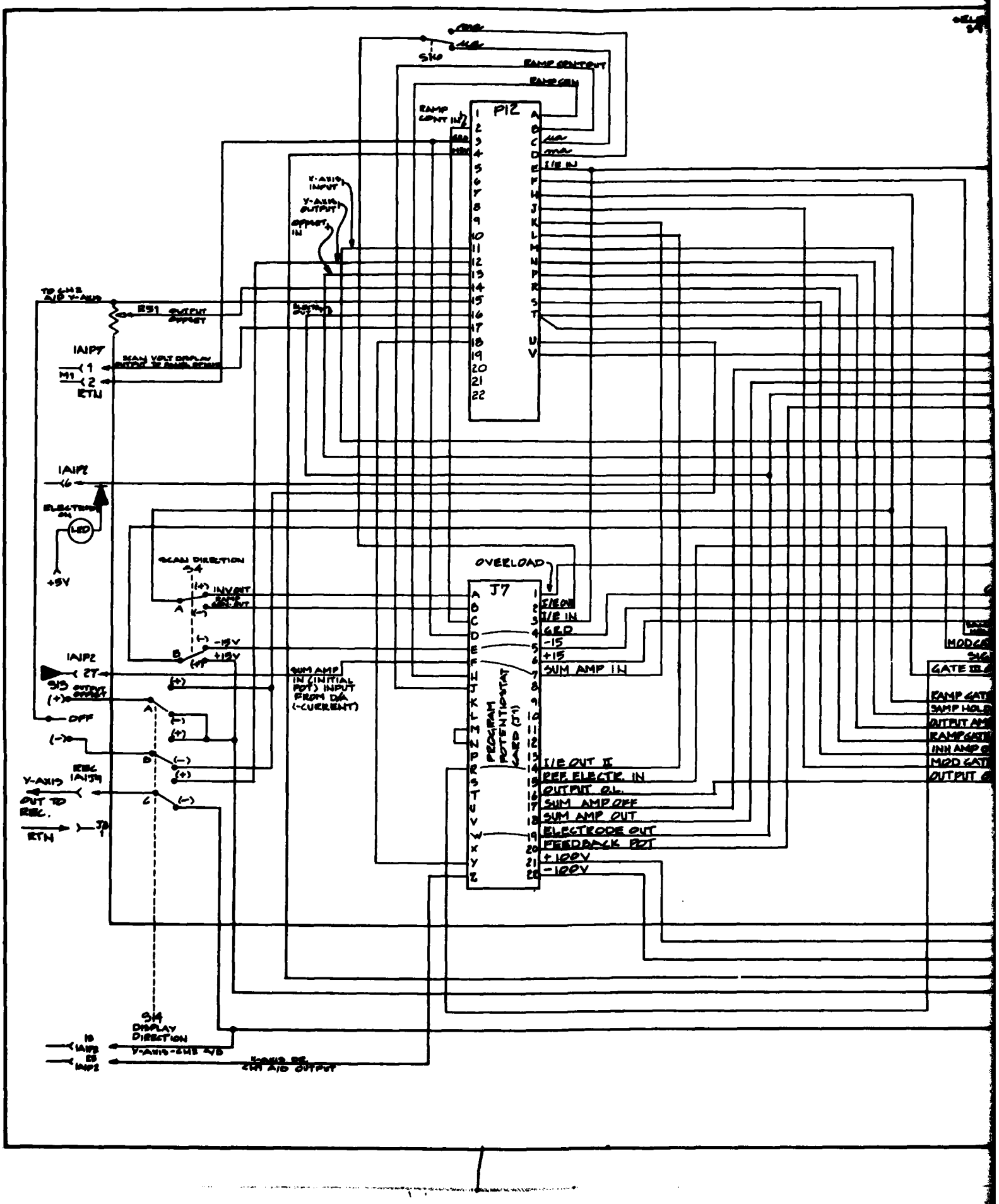
SEA WATER ANALYZER
EXTRA BOARD 2

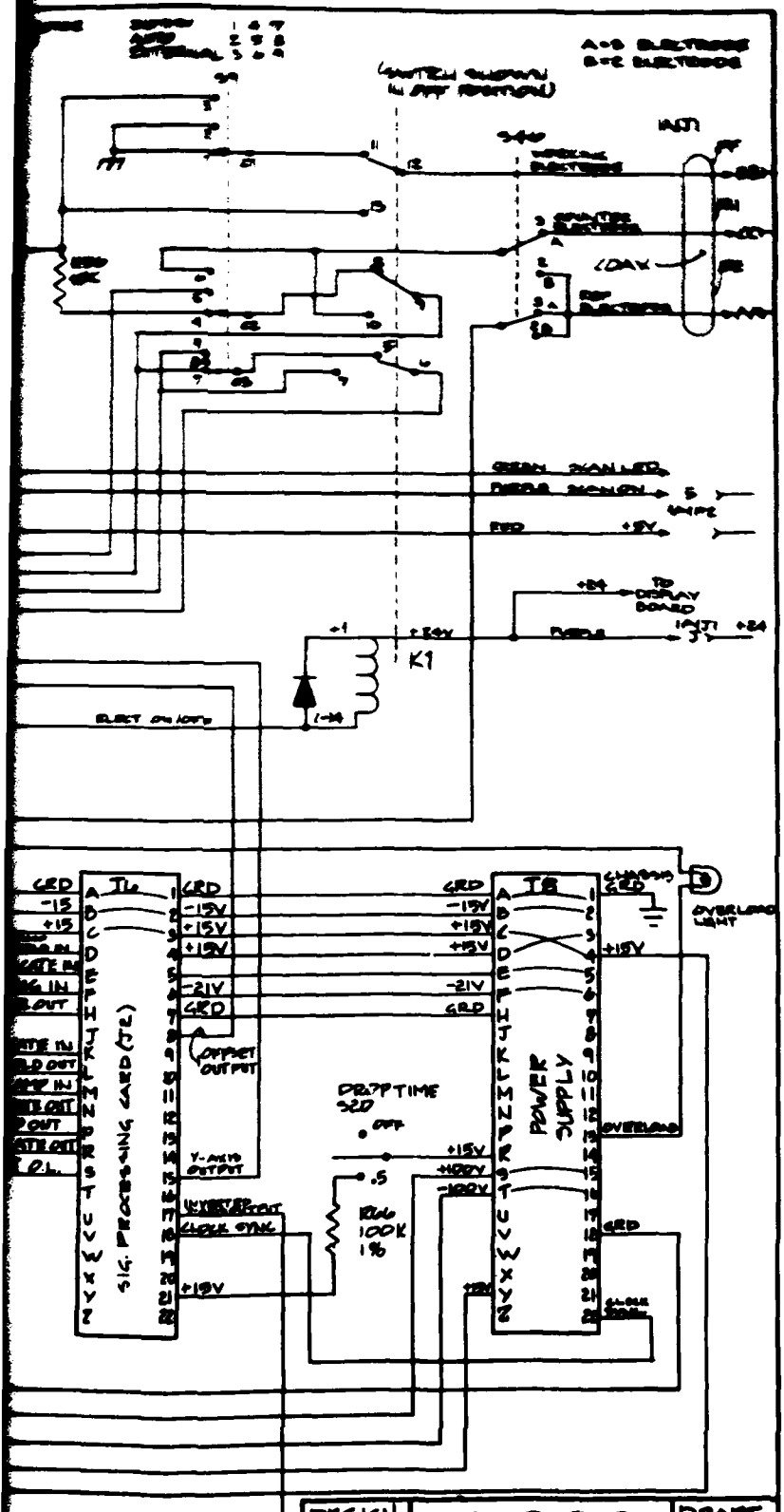
SHT 1 OF 1


PIA-5
ADDR: 7Ø2Ø - 7Ø23
J1
19 ENP5
14 IRQ
4 DØ
24 D1
5 D2
25 D3
6 D4
26 D5
7 D6
27 D7
16 RSØ
36 RS1
39 ENP6
15 R/W
34 ØZ
35 RESET
+5
GRD
20
22 CSØ
24 CS1
23 CS2
34 RESET
25 EN
21 R/W
33 DØ
32 D1
31 D2
30 D3
29 D4
28 D5
27 D6
26 D7
36 RSØ
35 RS1
5
6820
PAØ
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PBØ
PB1
PB2
PB3
PB4
PB5
PB6
PB7
CA1
CA2
CB1
CB2
IRQA IRQB
38 37
1K RES NET
2.2K
2.7K
74504
PIA-6
ADDR: 7Ø24 - 7Ø28
J1
+5 1, 21
GRD 20, 40

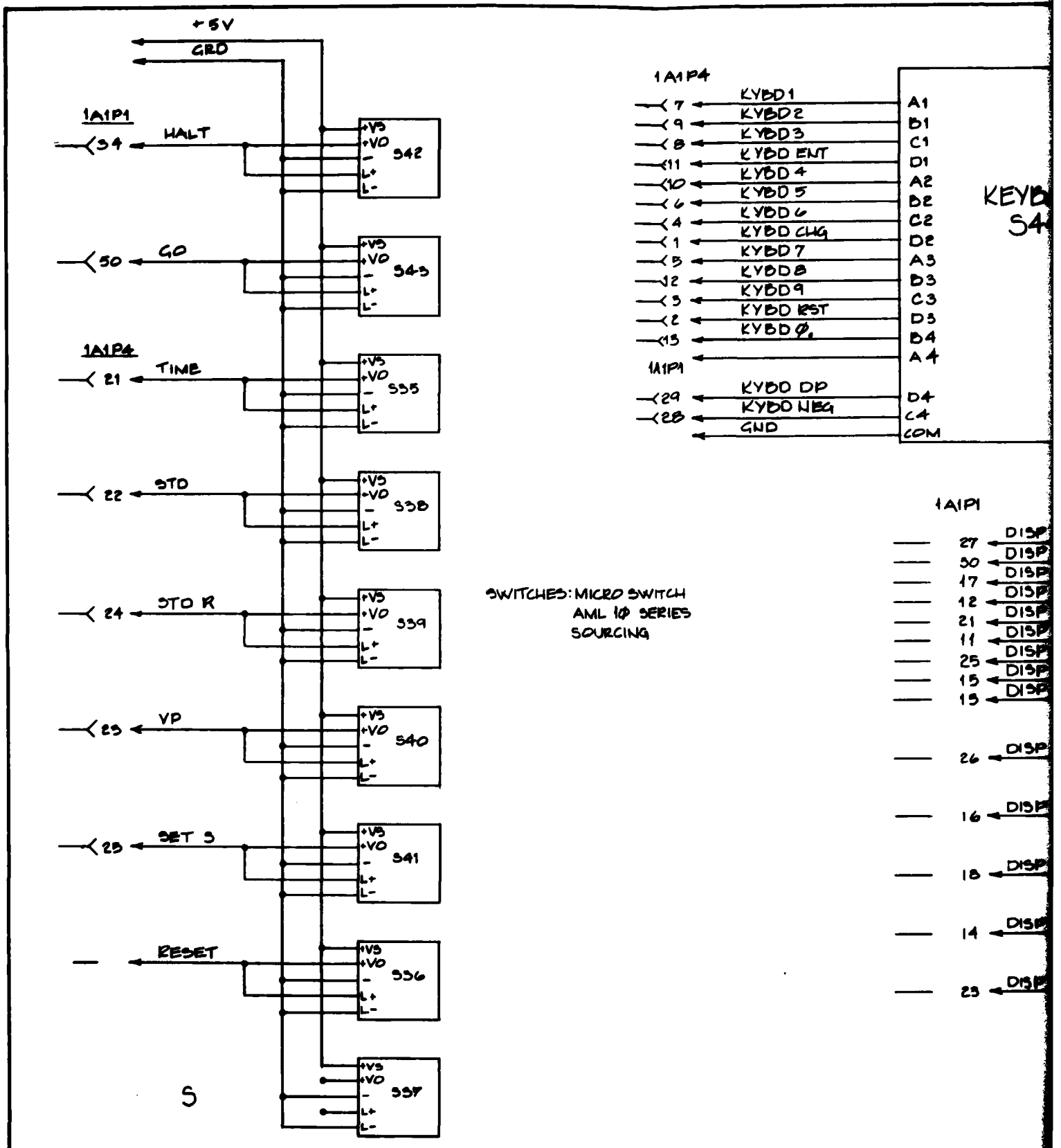


DESIGN C. CLAVELL	NOSC	DRAFT EMK.
DRW# CC-13	9-7-79.	REV# 1
SEA WATER ANALYZER EXTRA BOARD 3	SHT 1 OF 1	


S16
RAMP CONT OUT
RAMP GEN
P12
RAMP CONT IN
I/E IN
Y-AXIS INPUT
Y-AXIS OUTPUT
OFFSET IN
TO CH2 A/D Y-AXIS
R31
OUTPUT OFFSET
SCAN VOLT DISPLAY OUTPUT TO SCAN SCREEN
1A1P7
M1
1
2
RTN
1A1P2
6
ELECTRODE ON
LED
+5V
SCAN DIRECTION S4
INVERT
SCAN DIRECT
-15V
+15V
A
B
1A1P2
27
S13 OUTPUT OFFSET
(+)
OFF
(-)
REC 1A1J9
Y-AXIS OUT TO REC.
RTN
J8
1
S14 DISPLAY DIRECTION
Y-AXIS CH2 A/D
10
1A1P2
25
SCOPE CH2 A/D OUTPUT
OVERLOAD
J7
I/E ON
I/E IN
GRD
-15
+15
SUM AMP IN
SUM AMP IN (INITIAL POT) INPUT FROM D/A (-CURRENT)
PROGRAM POTENTIOSTAT CARD (J1)
I/E OUT II
REF. ELECTR. IN
OUTPUT G.L.
SUM AMP OFF
SUM AMP OUT
ELECTRODE OUT
FEEDBACK POT
+100V
-100V
MOD GATE
GATE
RAMP GATE
SAMP HOLD
OUTPUT AMP
RAMP GATE
INH AMP
MOD GATE
OUTPUT

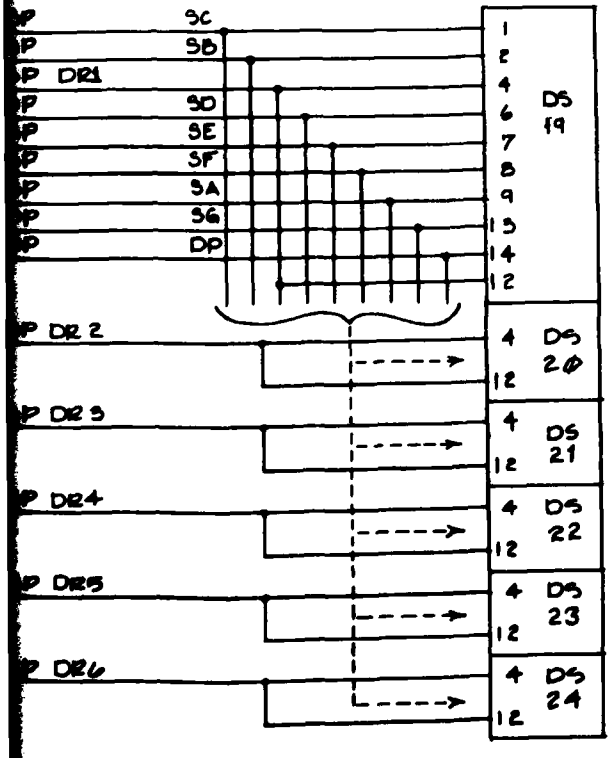


DESIGN C. CLAVELL	NOSC	DRAFT EMK
DRW# CC-14	9-7-79	REV# 1
POTENTIOSTAT BACKPLANE		
		SHT 1 OF 1


+5V
GRD
1A1P1
34 HALT
50 GO
1A1P4
21 TIME
22 STO
24 STO R
23 VP
25 SET 3
RESET
S42
S43
S35
S38
S39
S40
S41
S36
S37
S
1A1P4
7 KYBD1
9 KYBD2
8 KYBD3
11 KYBD ENT
10 KYBD 4
6 KYBD 5
4 KYBD 6
1 KYBD CHG
5 KYBD 7
12 KYBD 8
3 KYBD 9
2 KYBD RST
15 KYBD Ø
A1
B1
C1
D1
A2
B2
C2
D2
A3
B3
C3
D3
B4
A4
KEYB
S4
1A1P1
29 KYBD DP
28 KYBD NEG
GND
D4
C4
COM
SWITCHES: MICRO SWITCH
AML 10 SERIES
SOURCING
1A1P1
27 DISP
30 DISP
17 DISP
12 DISP
21 DISP
11 DISP
25 DISP
15 DISP
19
26 DISP
16 DISP
18 DISP
14 DISP
23 DISP

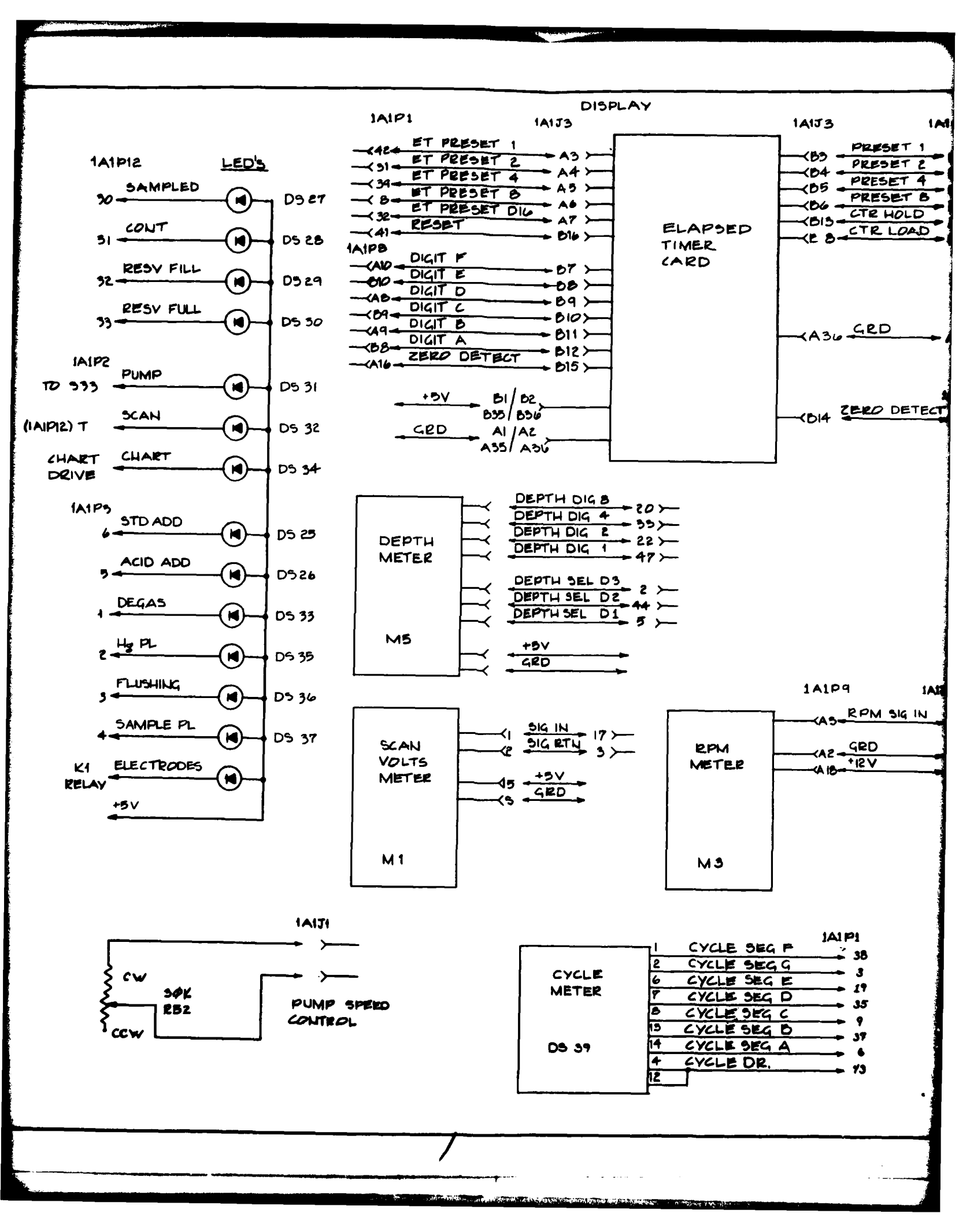
OARD
44

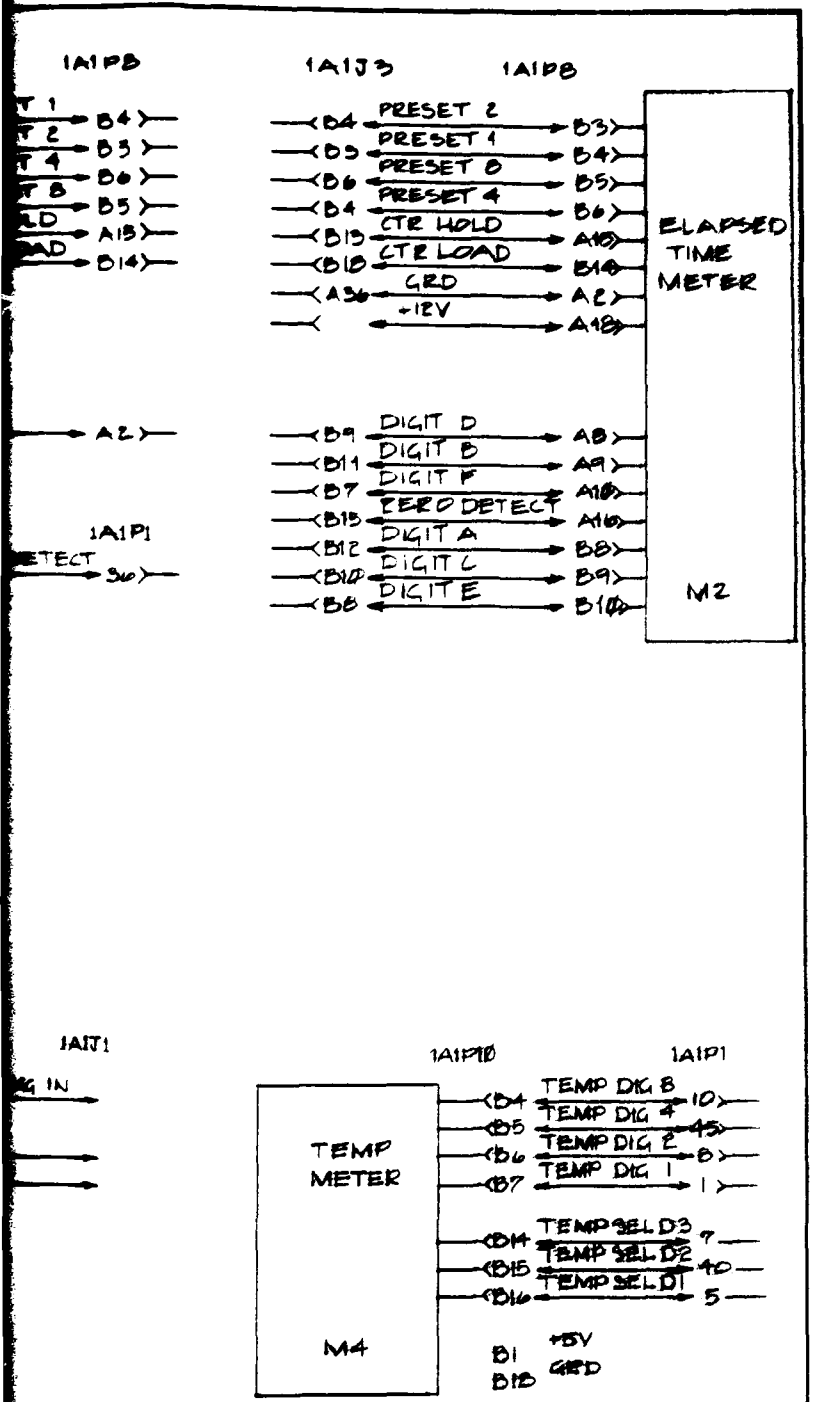
MAN 3640
7-SEG LED'S



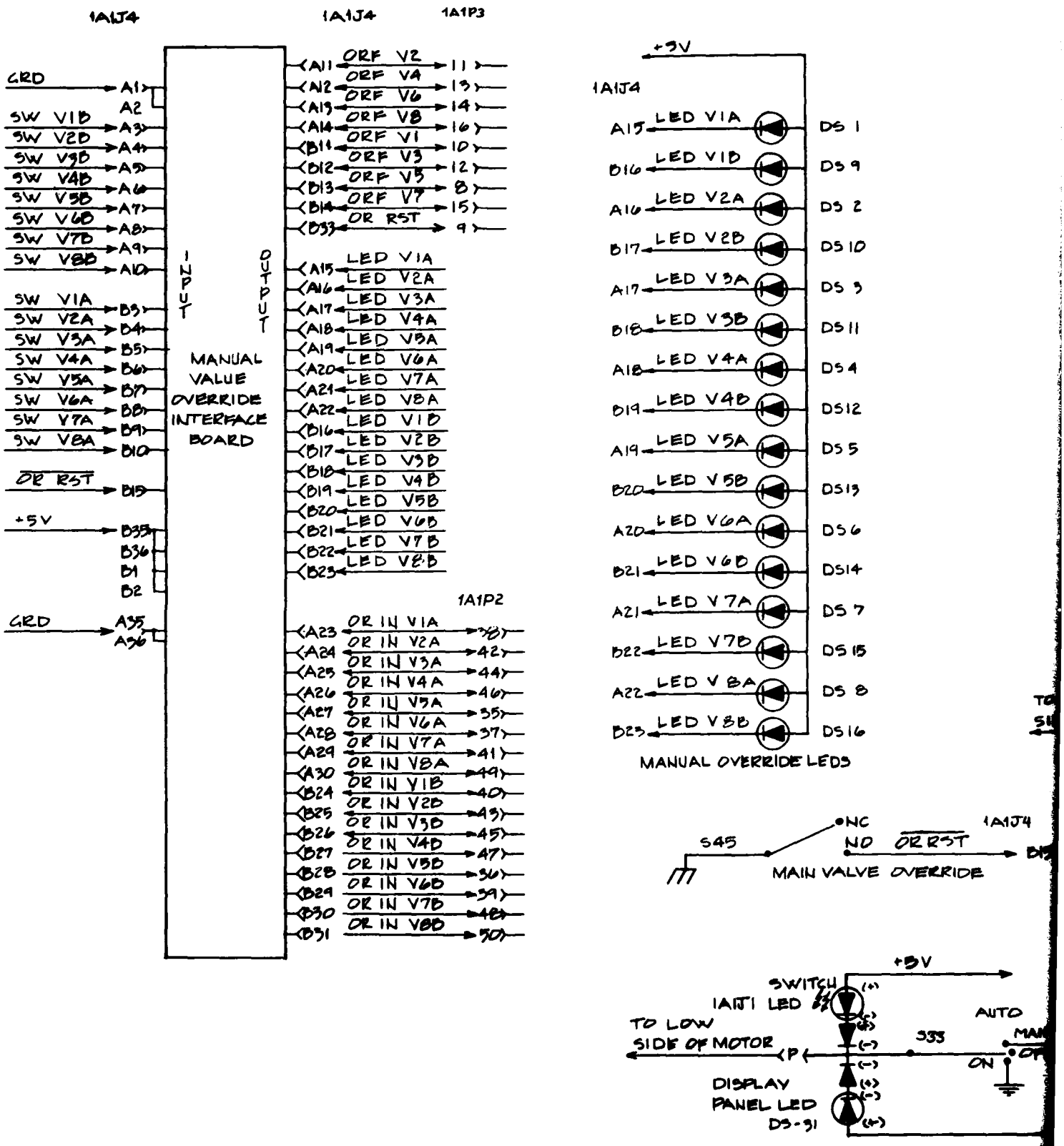
DATA DISPLAY

DESIGN C.CLAYELL	NOSC	DRAFT EMK
DRW# CC-15	9-7-79	REV.# 1
PROGRAM CONTROL & SELECT SWITCHES FRONT PANEL I/O	SHT 1 OF 1	

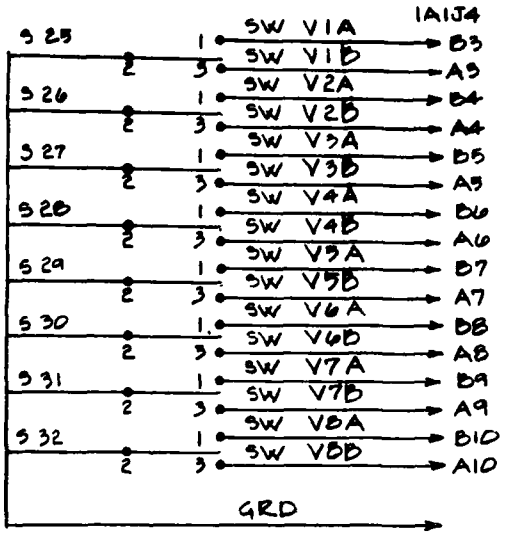

DISPLAY
1A1P1
1A1J3
1A1J3
1A1P12
LED'S
SAMPLED
CONT
RESV FILL
RESV FULL
1A1P2
PUMP
TO 333
SCAN
(1A1P12) T
CHART
CHART DRIVE
1A1P3
STD ADD
ACID ADD
DEGAS
Hg PL
FLUSHING
SAMPLE PL
K1 RELAY
ELECTRODES
+5V
DS 27
DS 28
DS 29
DS 30
DS 31
DS 32
DS 34
DS 25
DS 26
DS 33
DS 35
DS 36
DS 37
ET PRESET 1
ET PRESET 2
ET PRESET 4
ET PRESET 8
ET PRESET D16
RESET
1A1P8
DIGIT F
DIGIT E
DIGIT D
DIGIT C
DIGIT B
DIGIT A
ZERO DETECT
+5V
GRD
ELAPSED TIMER CARD
PRESET 1
PRESET 2
PRESET 4
PRESET 8
CTR HOLD
CTR LOAD
GRD
ZERO DETECT
DEPTH METER
M5
DEPTH DIG 8
DEPTH DIG 4
DEPTH DIG 2
DEPTH DIG 1
DEPTH SEL D3
DEPTH SEL D2
DEPTH SEL D1
+5V
GRD
SCAN VOLTS METER
M1
SIG IN
SIG RTN
+5V
GRD
1A1P9
RPM METER
M3
RPM SIG IN
GRD
+12V
1A1J1
CW
30K
RB2
CCW
PUMP SPEED CONTROL
CYCLE METER
DS 39
CYCLE SEG F
CYCLE SEG G
CYCLE SEG E
CYCLE SEG D
CYCLE SEG C
CYCLE SEG B
CYCLE SEG A
CYCLE DR.
1A1P1

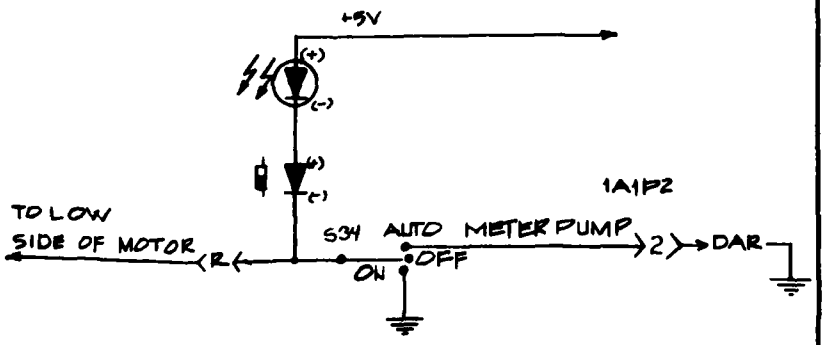


DESIGN C. CAMPBELL	NOSC	DRAFT EMK.
DRW# CC-16	9-7-77	REV# 1
FRONT PANEL I/O DISPLAY		SHT 1 OF 1


1A1J4
1A1J4
1A1P3
GRD
A1
A2
SW V1B
A3
SW V2B
A4
SW V3B
A5
SW V4B
A6
SW V5B
A7
SW V6B
A8
SW V7B
A9
SW V8B
A10
SW V1A
B3
SW V2A
B4
SW V3A
B5
SW V4A
B6
SW V5A
B7
SW V6A
B8
SW V7A
B9
SW V8A
B10
OR RST
B15
+5V
B35
B36
B1
B2
GRD
A35
A36
INPUT
OUTPUT
MANUAL VALUE OVERRIDE INTERFACE BOARD
A11 ORF V2 11
A12 ORF V4 13
A13 ORF V6 14
A14 ORF V8 16
B11 ORF V1 10
B12 ORF V3 12
B13 ORF V5 8
B14 ORF V7 15
B33 OR RST 9
A15 LED V1A
A16 LED V2A
A17 LED V3A
A18 LED V4A
A19 LED V5A
A20 LED V6A
A21 LED V7A
A22 LED V8A
B16 LED V1B
B17 LED V2B
B18 LED V3B
B19 LED V4B
B20 LED V5B
B21 LED V6B
B22 LED V7B
B23 LED V8B
1A1P2
A23 OR IN V1A 38
A24 OR IN V2A 42
A25 OR IN V3A 44
A26 OR IN V4A 46
A27 OR IN V5A 35
A28 OR IN V6A 37
A29 OR IN V7A 41
A30 OR IN V8A 49
B24 OR IN V1B 40
B25 OR IN V2B 43
B26 OR IN V3B 45
B27 OR IN V4B 47
B28 OR IN V5B 36
B29 OR IN V6B 39
B30 OR IN V7B 48
B31 OR IN V8B 50
+5V
1A1J4
A15 LED V1A DS 1
B16 LED V1B DS 9
A16 LED V2A DS 2
B17 LED V2B DS 10
A17 LED V3A DS 3
B18 LED V3B DS 11
A18 LED V4A DS 4
B19 LED V4B DS 12
A19 LED V5A DS 5
B20 LED V5B DS 13
A20 LED V6A DS 6
B21 LED V6B DS 14
A21 LED V7A DS 7
B22 LED V7B DS 15
A22 LED V8A DS 8
B23 LED V8B DS 16
MANUAL OVERRIDE LEDS
S45
NC
NO
OR RST
1A1J4
MAIN VALVE OVERRIDE
+5V
SWITCH
1A1J1 LED
TO LOW SIDE OF MOTOR
P
335
AUTO
ON
DISPLAY PANEL LED DS-31

MANUAL OVERRIDE SWITCHES





B13

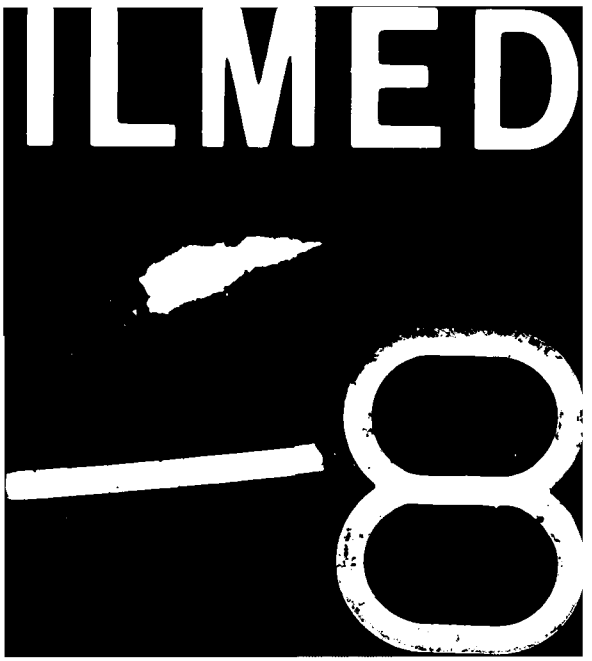
1A1P2

IN PUMP
FF
1
DAR

+5V

DESIGN C. CLARK	NOSC	DRAFT EHK.
DRW# CC-17	9-7-79.	REV# 1
MANUAL OVERRIDE FRONT PANEL I/O		SHT 1 OF 1

2


ILMED
8