FILE COPY

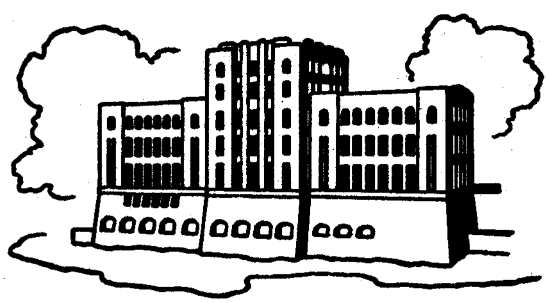
COMPUTER RECOGNITION OF CONTRACTIONS IN THE SMALL INTESTINE

by

Robert B. J. Singerman and John R. Glover

Sponsored
by
National Institutes of Health
Grant Numbers
AM08901, AM5390, and AM20547
and
National Science Foundation
Grant Number GK 16285

PLEASE DO NOT REMOVE



IIHR Report No. 134

Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City, Iowa

September 1971

COMPUTER RECOGNITION OF CONTRACTIONS IN THE SMALL INTESTINE

by

Robert B. J. Singerman and John R. Glover

Sponsored
by
National Institutes of Health
Grant Numbers
AM08901, AM5390, and AM20547
and
National Science Foundation
Grant Number GK 16285

IIHR Report No. 134

Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City, Iowa

September 1971

ABSTRACT of

COMPUTER RECOGNITION OF CONTRACTIONS IN THE SMALL INTESTINE

by

Robert B. J. Singerman and John R. Glover

A computer program for processing electrical signals representing contractions in the small intestine has been developed. The program has been written specifically for an IBM 1801 Data Acquisition System having at least seven channels of analog input and one channel of the electronic-contact output. The processing techniques for identifying contractions are not computer dependent, although the data acquisition phase of the program is.

The program recognizes contractions and is able to distinguish them from movement artifacts and electrical noise. Changes in base-line pressure are accounted for by using the current value of the minimum signal in the computations for identifying a contraction. Once a contraction is identified, the time interval since the detection of the last contraction is recorded along with the corresponding channel number. After six contractions have been identified, the time intervals and corresponding channel numbers are punched to create a permanent record. The technique can be applied to the processing of similar records (i.e., bell shaped) generated by isolated events.

ACKNOWLEDGEMENTS

Grateful acknowledgement is extended to Dr. James Christensen and Dr. Norman Weisbrodt of the Gastroenterology Research Laboratory, Veterans Administration Hospital, Department of Internal Medicine, College of Medicine and to Dr. Enzo Macagno of the Institute of Hydraulic Research, College of Engineering, for their suggestions and guidance in this phase of the study of the statistics of contractions in the human duodenum. This work was supported by Research Grant AM08901, Training Grant AM5390, and Career Development Award AM20547 from the National Institute of Arthritis and Metabolic Diseases, National Institutes of Health. The expansion of the computer facilities needed for this study was provided by equipment grant GK 16285, National Science Foundation.

TABLE OF CONTENTS

LIST OF FIGURES

Page

COMPUTER RECOGNITION OF CONTRACTIONS IN THE SMALL INTESTINE

I. INTRODUCTION

When one undertakes the study of the occurrence of a physiological event, he is generally confronted with a realization that statistical approaches must be incorporated into the analytical procedures in order to better understand the process. With current-day computer technology, the application of statistical procedures is usually very straightforward once the data have been collected. It is often the situation, though, that data which represent a physiological event are recognized and defined by an experienced "eye." To remove the scientist from this mundane role of processing thousands of occurrences of the event, it is desirable to program a computer with a recognition scheme with constant judgement that gives comparable results. For the study reported herein, it was necessary to develop a pattern-recognition procedure which recognized the occurrence of a contraction in the small intestine.

A well-established method for detecting contractions in the small intestine has been employed and described by Christensen, Glover, Macagno, Singerman, Weisbrodt (1971). The complete system for detecting, recording, and processing the signals representing contractions in the intestine is shown schematically in figure 1. The transducer, oscillograph, and tape recorder are connected without modifications (that is, the units are connected in the usual and anticipated manner). The system includes the tape recorder because the patient-study facilities and the computer facilities are located in different laboratories. The recorder serves no purpose other than to transfer the electrical signals between the two facilities.

Figure 2 is a photograph of the signal reproduced by the tape recorder. It shows the shape and quality of the signal representing a contraction in the intestine. Most of the noise observable in the figure originates within the tape recorder. Base-line drift, which is attributable to ambient pressure changes in the patient, and the noise presented the main difficulties in processing the records.

The computer program was designed to recognize contractions and to distinguish these from movement artifacts and electrical noise, and to compute the time interval between successive contractions on any of seven different channels. Figure 2 shows a single-channel record and the results of a single-channel version of the seven-channel program. The pulses are generated by the computer upon recognition of a contraction. Pulses for each channel of the seven-channel program are not generated because of an inadequate number of recording channels. Recognition of contractions is achieved by a moving-summation technique for analyzing the amplitude of the voltages representing a contraction. With this technique, a summation of a block of the ordinants of the trace shown in figure 2 is displaced systematically while its value is being constantly observed. As the summation advances through time it reaches extreme values: maximum values are identified as pressure peaks, and minimum values constitute basal pressures between peaks. The amplitude of the signal is sampled every ten milliseconds by the analog-to-digital converter (ADC) in the computer and stored in memory. A block of thirty samples is kept with the newest sample replacing the oldest sample every ten milliseconds. The sum of the thirty samples is proportional to the average of the thirty samples and to the area under the curve of the signal over a time interval of 0.30 second. Two tests are used to aid in the identification of peaks. Each maximum summation is compared to the preceding minimum value and required to be a constant amount greater. Each summation is also required to decline, after passing a peak value, by a constant amount. The first test is used to reject artifacts of low amplitudes such as those due to respiration. The second test is used to reject slight irregularities or notches that appear in some peaks. When both tests are satisfied for any one peak, it is counted as a contraction. The time since the last contraction is recorded along with the channel number. The computer output is a series of times representing intervals between successive contractions in the duodenum along with corresponding channel numbers. This method defines each peak with respect to the preceding baseline. Since the base line is calculated each time as a minimum value after a contraction is recognized, and since the next peak is identified as a difference between that minimum value and the succeeding maximum value, the technique compensates for base-line shift.

Specific aspects of the program are best understood by considering the program flow chart rather than the Assembler language program directly (see Appendix A). The flow chart is broken into segments shown in figures 3 to 11, and these segments are discussed in the following sections of the report. As indicated the program is written in the Assembler language, and each part is identified with its corresponding segment of the flow chart. Program statements corresponding to the flow chart are not discussed. Figure 3 is a generalized flow chart of the main body of the program and will be discussed prior to the specific sections of the program.

II. GENERALIZED FLOW CHART

The first step indicated in the flow chart shown in figure 3 is one of program initiation. This phase of the program establishes addresses, constants, and other initial conditions necessary for proper execution of the program. The initiation indicated by the instructions in the program is determined by the hardware configuration of the Institute's IBM 1801. As a result, this phase of the program must be tailored to each computer system because of the numerous possibilities for defining the interrupt structure of the IBM 1801. The end of this section of the program is indicated when the message START TAPE RECORDER THEN PUSH START is printed. This pause is to permit time for the operator to start the tape recorder.

Once the start button is pressed, timer B is started and index register 2 is set equal to -7 to establish the loop for the contraction tests of the seven channels. The computer then waits ten milliseconds for the timer interrupt which causes the timer-interrupt routine to be executed. Another 10-millisecond time interval is established and seven new data points are collected and stored for evaluation in the contraction-test section of the program by the timer-interrupt routine. After the timer-interrupt routine, all seven channels are tested for a contraction and the appropriate information is recorded. The program then returns to WAIT 7 for the remainder of the 10-millisecond time interval to expire.

III. PROGRAM INITIATION

As was indicated above, this section (see figure 4) of the program must be tailored to each 1801 installation because of the numerous possibilities of the interrupt configuration of the system. The interrupt level which must be modified for operation of this program is the interrupt level assigned to the interval timers. To transfer control of the timer-interrupt level from IBM's Time-Sharing Executive System to this program, the timer must first be stopped to eliminate an interrupt which would be serviced by a TSX subroutine. Once the timer is stopped, the contents of the core location containing the address of the TSX timer-interrupt routine must be saved and replaced by the address of the timer routine in this program. This address is used by the forced BSI (branch and store instruction register) indirect instruction which is executed when the timer becomes equal to zero. The original address contained in the modified core location must be restored before control of the timer is transferred back to TSX. Once these modifications have been completed the contents of the timer can be set to the value corresponding to the desired time interval and the timer started.

Two of the three constants entered through the Data Entry Switches on the computer console during the initiation phase define the values which must be satisfied before a contraction is recognized. These two double-precision constants (CON1 and CON2) are determined by comparing the strip-chart record showing a series of contractions with the result of the computer recognition program (that is, the output from this program). These constants are fixed once they are determined unless the sensitivity of the transducer-recorder system changes. The quantity CON1 defines the amount the summation must be above the noise level of the record, and CON2 defines the amount the summation must drop below the maximum summation before a contraction is recognized. Additional details concerning these two constants will be presented in the section describing the contraction tests.

Constant CON3 is the value of the time interval between samples made by the ADC of the seven analog channels. Normally, the value of this constant is minus 10 milliseconds.

The final phase of the initiation section defines various constants for the contraction testing section, clears the core locations for the seven blocks of 30 samples, and types the message which terminates this section.

IV. TIMER-INTERRUPT ROUTINE

The timer-interrupt routine (see figure 5) is entered by an automatic branch from normal program sequence. The branch is executed when the interval timer has concluded the recording of a selected timer interval. The automatic branching operation stores the address of the next sequential instruction in the main program to be executed in location TIMER and then executes the first instruction in the interrupt routine. Normally the contents of the A and Q registers, the index registers and machine status indicators are saved by the interrupt routine. Following execution of the interrupt routine, the contents of the A and Q registers, the index registers, and the machine status indicators are restored to their original values, and control is returned to the main program via a BSC I TIMER instruction. Thus no information is lost and the main program execution continues. However, if an IBM library subroutine is interrupted and a second IBM library subroutine is called, then information will be lost and the program will fail to function. Since this possibility exists for the program described herein, additional operations are necessary to prevent the loss of information vital for correct return to the interrupted library subroutine. To understand how information is lost it is necessary to examine the structure of a typical IBM library subroutine.

Immediately after branching to an IBM library subroutine, the contents of the A and Q registers, the index registers, and machine status indicators are saved by a call to another IBM library subroutine. The subroutine which performs these save operations is called by a BSI I 172 instruction and is called TVSAV. Exit from the IBM library subroutine is accomplished by using TVEXT and is called by a BSI I 173 instruction. This routine (TVSAV) stores the contents of the A and Q registers, the index registers, and the machine-status indicators and returns control to the calling program via a BSC (branch or skip on condition) indirect instruction to the address stored in low core by TVSAV. These subroutines produce the same results as the set of instructions described above for normal execution of an interrupt routine. They are in subroutine form, though, in order to eliminate unnecessary redundancy in all of the IBM library subroutines. It is thus seen, that the nesting of IBM subroutines permits the execution of the BSI I 172 instruction a second time before the BSI I 173 instruction (TVEXT) is executed. Therefore, steps must be taken which preserve the information saved by TVSAV prior to calling it a second time before TVEXT is called for the first time. A description of these steps follows below.

The first operation in the timer interrupt routine must be one which saves the contents of the A and Q registers. The contents of core location 55 must also be saved because location 55 contains the address of the BSC I instruction which provides the exit for TVEXT. To save the information stored by TVSAV as a result of a call by the first IBM subroutine it is necessary to call TVEXT. To prevent the return to the part of the program following the call to the original IBM subroutine, an address must be stored in location 55 prior to calling TVEXT. The address stored is simply the address of the instruction following the call to TVEXT. Once TVEXT has been executed, the registers and indicators must again be saved. After the execution of the second IBM library subroutine, the last set of registers and indicators saved are restored and TVSAV is called. The original address in location 55 is also restored for the call to TVEXT by the interrupted IBM subroutine. When these steps have been accomplished, the events necessary to permit the return to an interrupted IBM subroutine are completed.

The remaining aspects of the timer-interrupt routine are the resetting of timer B, the acquisition of the seven new data points, and the insertion of them into the respective blocks of 30 words. Because of the insertion process for placing new data in the block of 30 words, addresses for the insertion operations are restored to their original values after every 30 executions of the timer-interrupt routine.

The timer-interrupt routine also contains an exit test so that control can be returned to the system. If an exit is indicated the program branches to the exit routine. If an exit test is not indicated, the registers are restored and the timer-interrupt routine exits to the main program by the BSC I TIMER instruction.

V. CONTRACTION TESTS

As indicated in Section I, INTRODUCTION, a moving-summation technique is used to decide if a contraction has occurred. A summation of 30 samples is used in this decision-making process to reduce the effect of high-frequency noise introduced by the tape recorder and other artifacts. The summation represents the average of the 30 samples and reduces the influence of the noise in the recognition process. The technique has better resolution in defining the peak than that provided by a single point evaluation, because the derivative of the signal is zero at the peak. Probably the easiest method for discussing the flow chart in figure 6 is to follow the sequence of events for one channel as a contraction is detected. The "N" in each of the variables SMAXN, TEMPN, BLOCN, and TESTN represents a subscript which corresponds to the channel being tested. Figure 7 provides a visual interpretation for the following discussion of the tests performed on the block of data.

For the positions of the block of samples prior to the position shown in figure 7, various tests will have been passed or failed. To illustrate the sequence of events, though, it is only necessary to assume that SUM is increasing. Since SUM is increasing, it is greater than TEMPN and MIN will be stored in TEMPN. The storing of MIN into TEMPN is necessary to insure repeatability for this first test once SUM starts to increase. SUM will be greater than SMAXN because the signal is increasing. Hence, SUM is stored in SMAXN and the next block of 30 samples, corresponding to the next channel, is tested. For this channel, the above sequence repeats after each timer interrupt until SUM is less than SMAXN. This condition corresponds to the sum of the block of 30 samples being less than the previous sum of the samples. However, SUM must decrease by the value of CON2 below SMAXN before it is concluded that the maximum amplitude has passed. Once the signal drops and SUM is less than SMAXN by the value of CON2, SMAXN is tested to see if it is greater than the previous minimum value of the signal plus CON1. Failure of this test indicates that the signal had the proper form, but lacked sufficient amplitude to be recognized as a contraction. If this latter test is passed, the channel number is recorded and conditions are re-established for the next sequence for testing this channel. A branch to COMMON is then executed. As long as the signal decreases, SUM is less than TEMPN and a new minimum value of the signal is recorded. This repeated computation for the minimum value of the summation of the samples corrects for base-line drift in the signal.

Once COMMON is executed, the contraction tests for the next channel are repeated. Generally all seven channels are tested before the next timer interrupt. On occasion, however, the timer-interrupt routine may interrupt this section because of delays introduced by the punching operation. When this occurs, one data point in each of the blocks of all channels not yet tested is lost. The loss of the data point can introduce an error in the recorded time between contractions of plus or minus one time interval. The design time interval for the program is ten milliseconds which may be increased but not decreased.

VI. NEXT

The contraction test for all seven channels is done within the same section of the main program. The program loops through this section seven times, once per each channel. Between each iteration of the contraction-test section, the program comes to NEXT (see figure 8). Here the addresses of the appropriate instructions in the contraction-test section are updated for use with the next channel to be tested. After all seven channels have been tested the addresses are restored to those used for channel 1.

VII. EXIT ROUTINE

The flow chart for the exit routine is shown in figure 9. The first operation in the routine is the stopping of timer B so that a timer interrupt will not occur. The address for interrupt level 00 is restored and the storage protect bit is set. Timer C is started so that complete control of the timers is returned to TSX. The final instruction is a CALL VIAQ, a subroutine which terminates the program.

VIII. COMMON BLOCK

This section of the program (see figure 10) is entered immediately following the recognition of a contraction. The first function of this section is to store the value of TIME (which contains the elapsed time since the previous contraction) and CHAN (which contains the channel number for which the current contraction occurred). These are stored in a block of twelve words. After the storing operation, the value of TIME is reset to zero and the addresses for the storage of TIME and CHAN are updated for the next identified contraction. A test to determine if 6 TIME's and 6 CHAN's have been stored in the block of 12 words is then done. If not, the program returns to the contraction-test section. If the block of 12 words is full, the program goes to the PUNCH section.

IX. PUNCH

The operation first executed by this section of the program (see figure 11) is the resetting of the addresses for storing TIME and CHAN. An ECO (electronic contact operate) pulse is generated and recorded on a strip-chart recorder along with the actual data for a visual record of when cards were punched. The 12 words are then converted to a form suitable for punching by the IBM subroutine BINDC, and then stored in the first 72 words of an 80-word block. A card number is then incremented by one, converted, and stored in the last six words. This block of 80 words is then punched on one card and the program goes to the section labelled NEXT.

X. CONCLUSION

The program described in the foregoing sections was devised with the following goals in mind: 1) to recognize peaks in an electrical signal assumed to correspond to small-bowel wall contraction; 2) to increase objectivity over that found with purely visual techniques; and 3) to increase the speed of recognizing time intervals between contractions over visual techniques. Peaks are recognized using a moving summation technique which defines both the local maximum value in the signal and the local minimum value of the signal. Since one of the test parameters used in defining peaks is based on the difference between the local maximum values and the local minimum values, any base-line drift is accounted for. Although this particular program does not make use of information regarding the relative amplitudes of the peaks, these data are available. The technique is not completely objective as the two contraction-tests parameters must still be chosen. Once selected, however, they remain constant and the contraction identification criteria remain unchanged during the analysis. A marked increase in capability for processing experimental data has resulted by programming the computer to process the data.

REFERENCES

[1.] Christensen, James, Glover, John R., Macagno, Enzo O., Singerman, Robert B., and Weisbrodt, Norman W., "Statistics of Contractions at a Point in the Human Duodenum," accepted Aug. 15, 1971, *American Journal of Physiology*, Am. 2 Physiol.

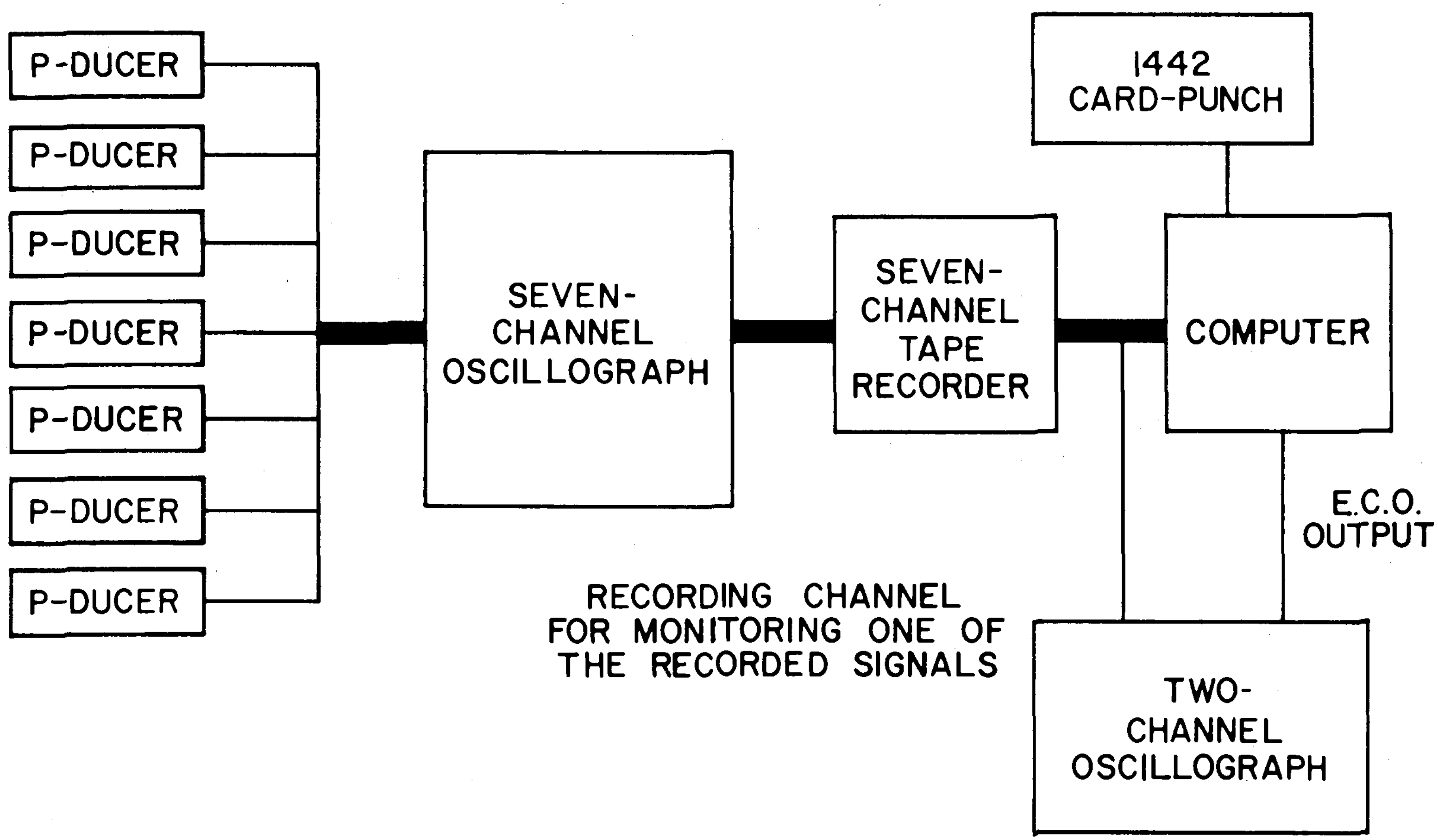


Figure 1. System for detecting, recording, and processing signals.

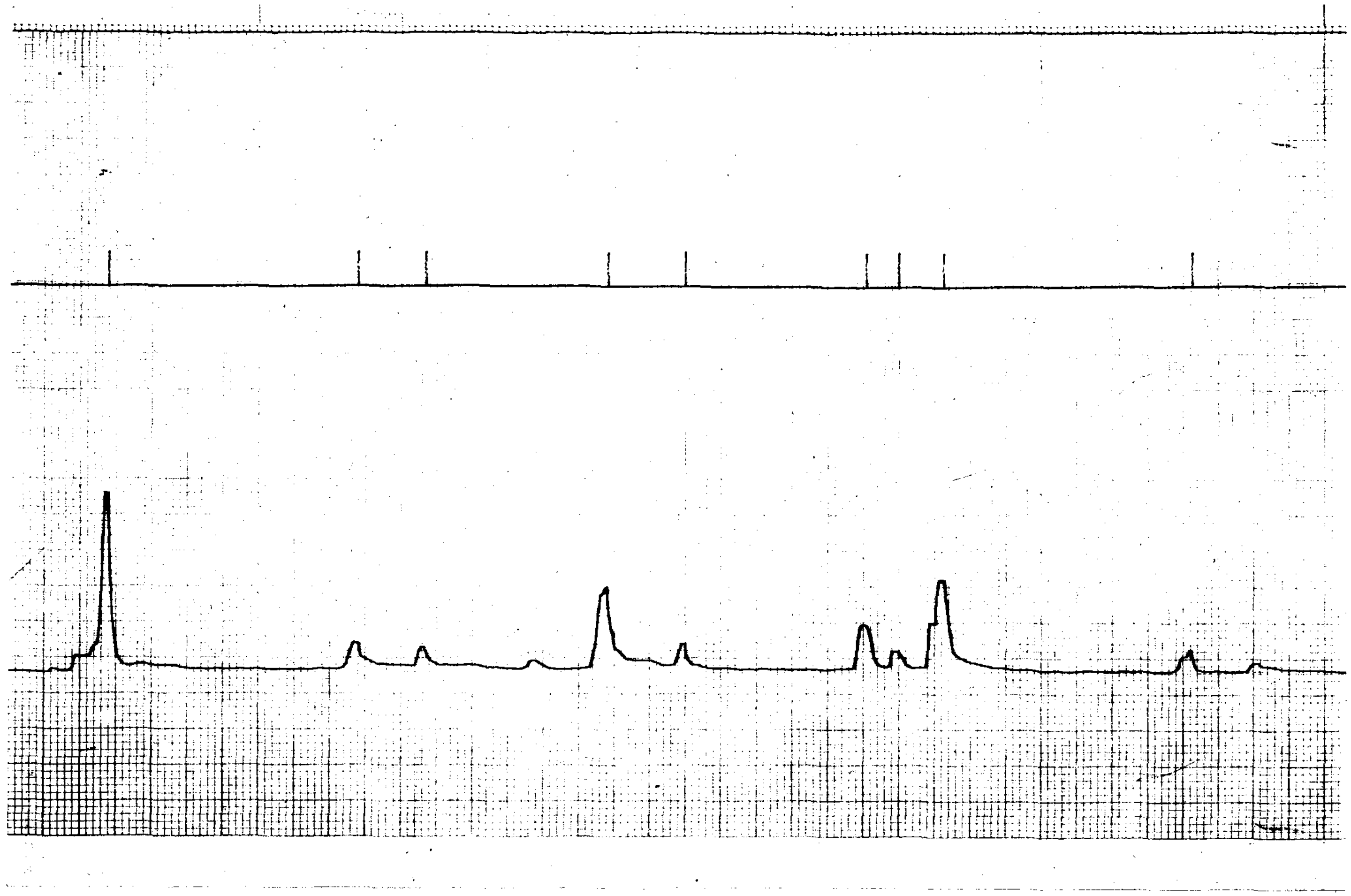


Figure 2. Reproduced signal for processing by the computer.

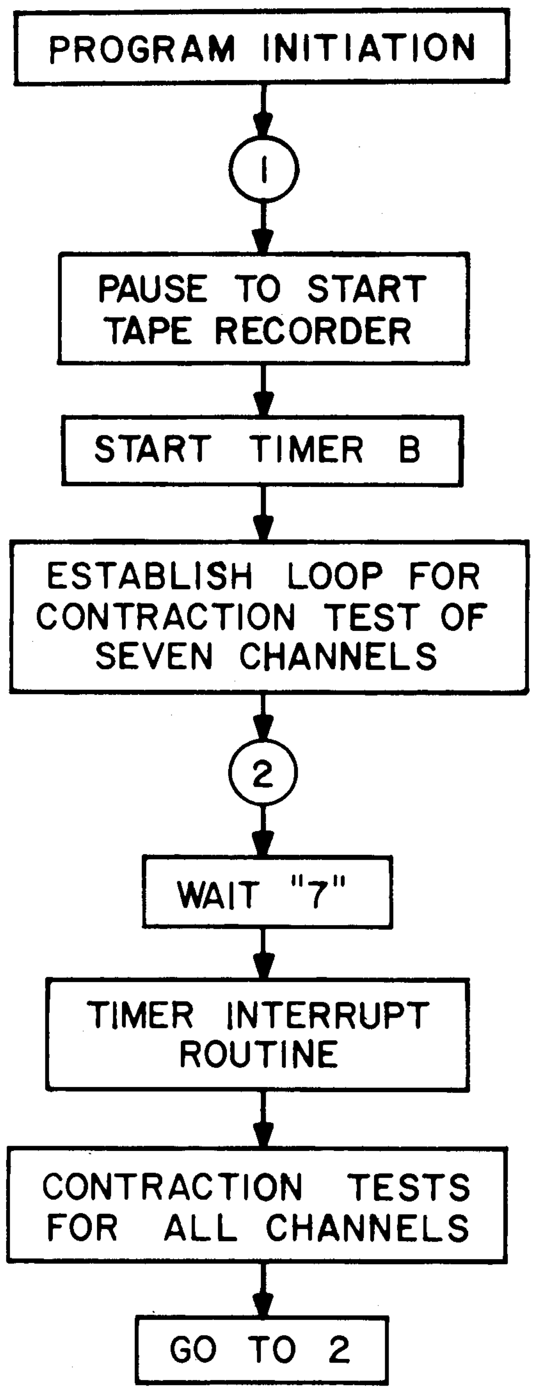


Figure 3. Generalized flow chart.

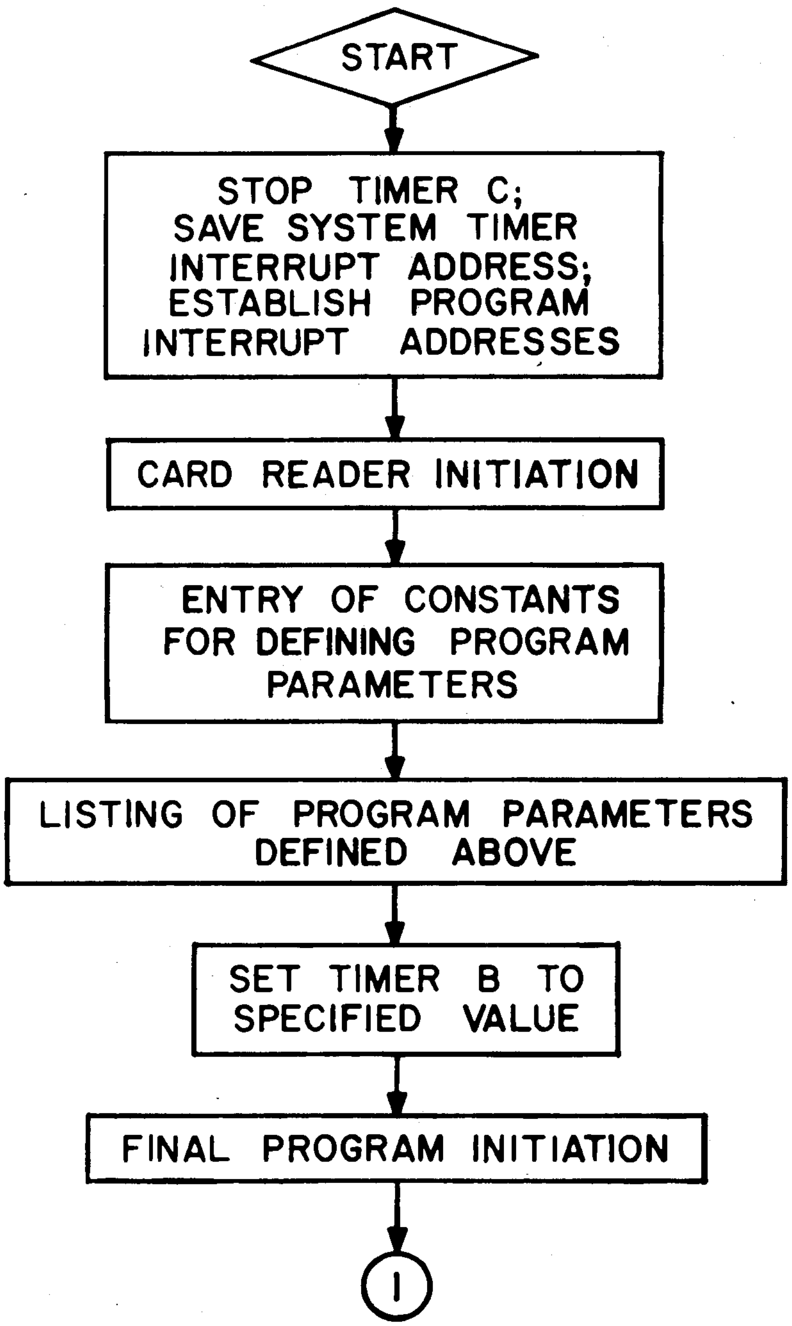


Figure 4. Flow chart of the program initiation.

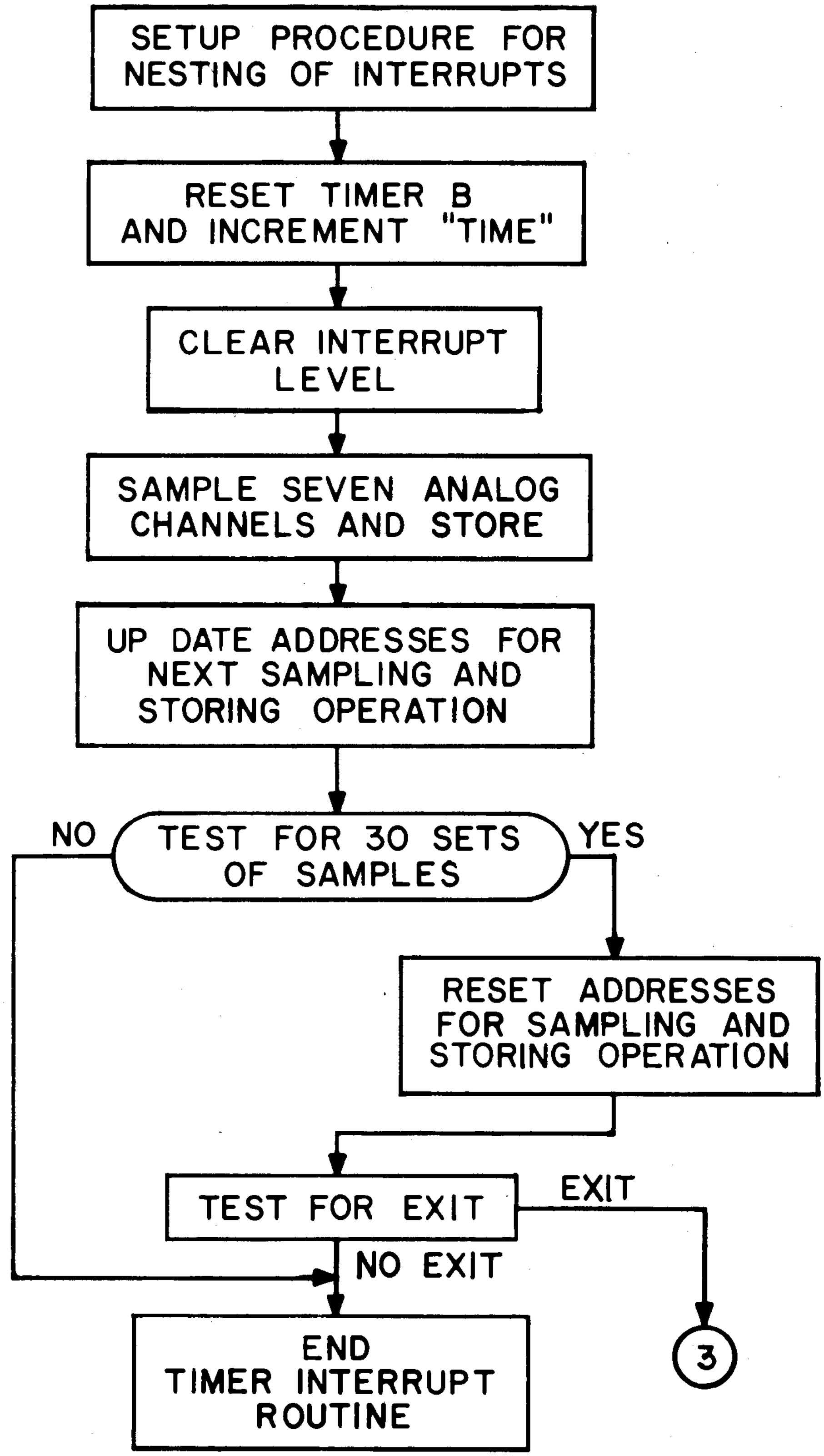


Figure 5. Flow chart of the timer-interrupt routine.

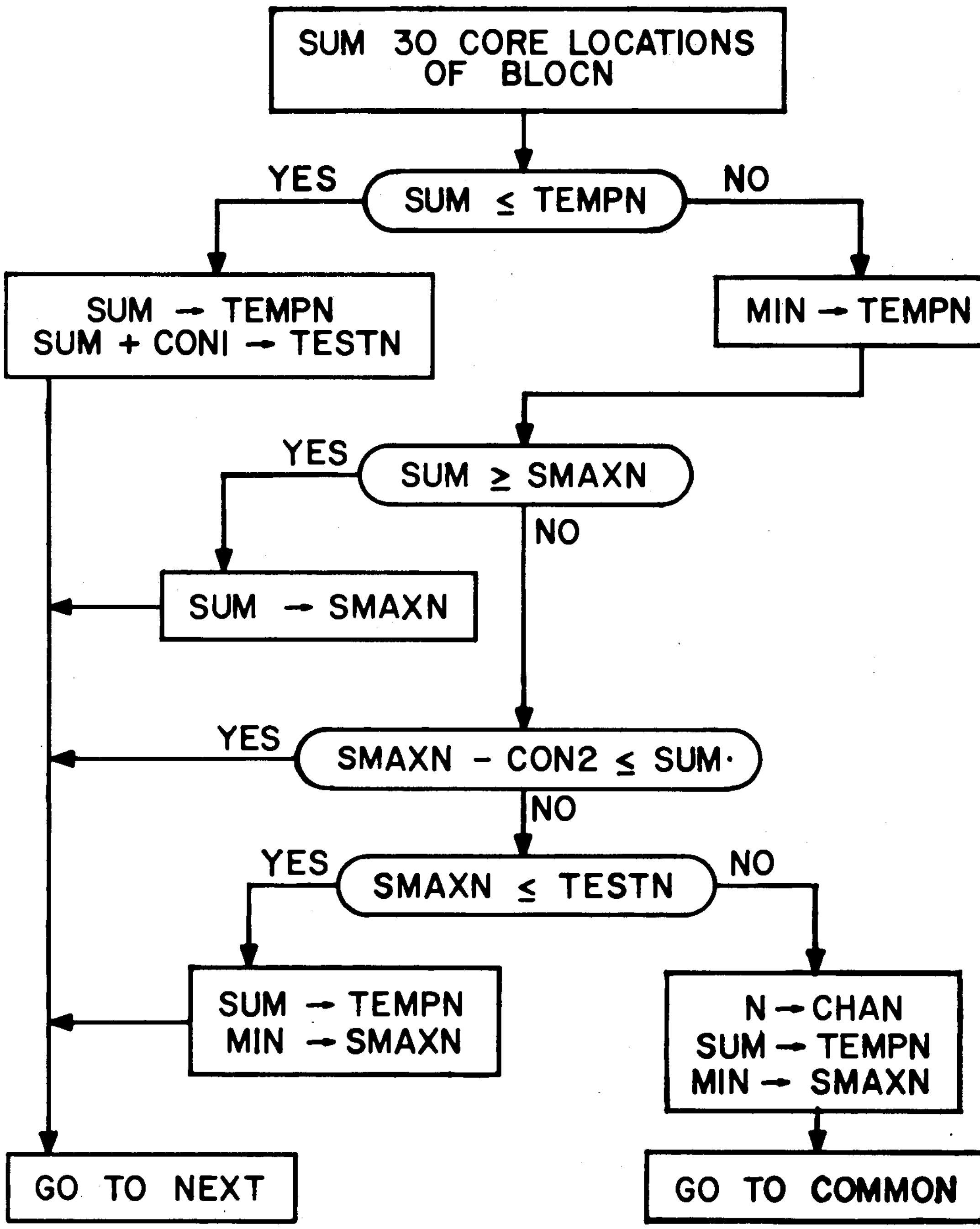


Figure 6. Flow chart of the contraction-tests routine.

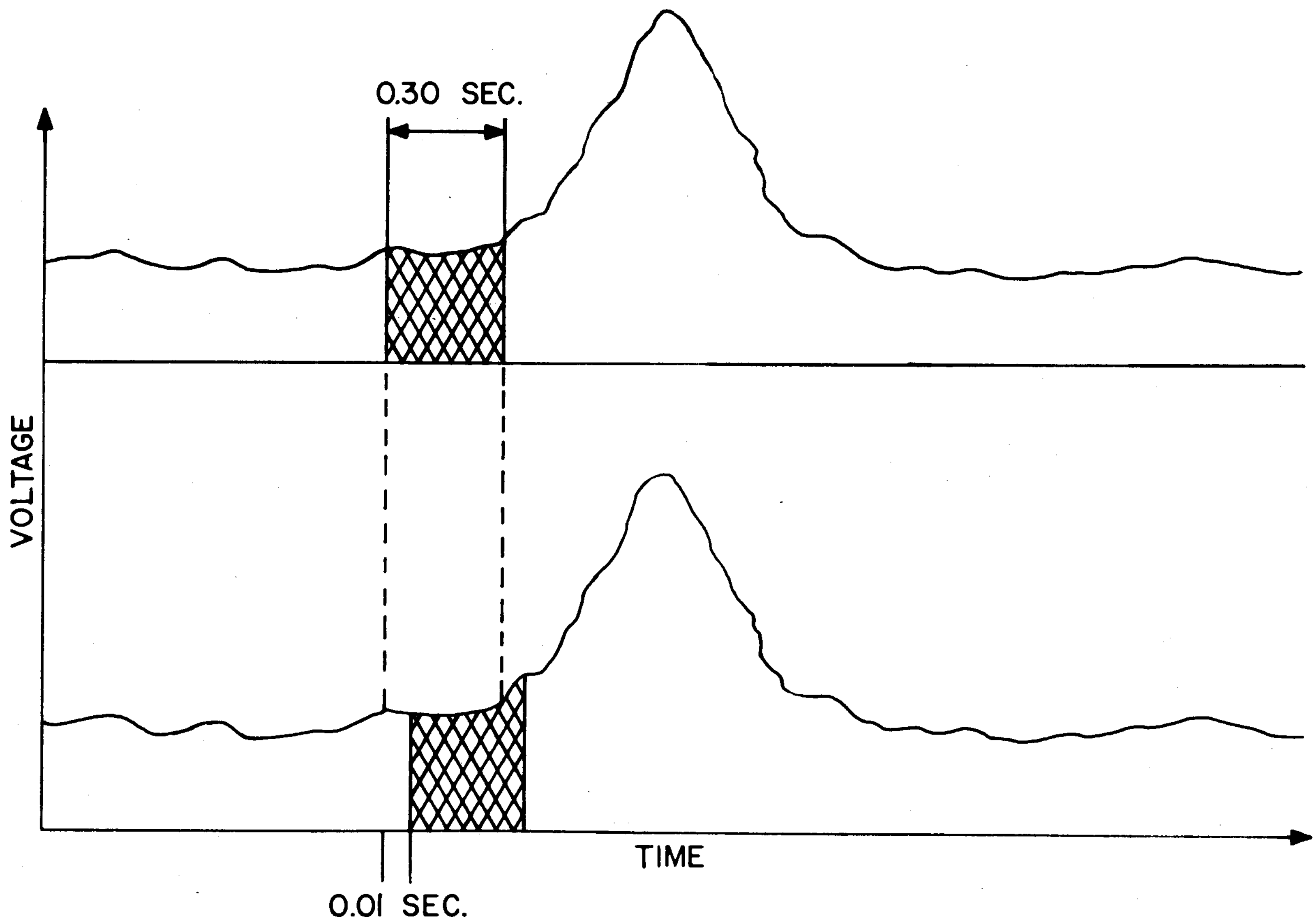


Figure 7. Schematic illustrating signal processing technique.

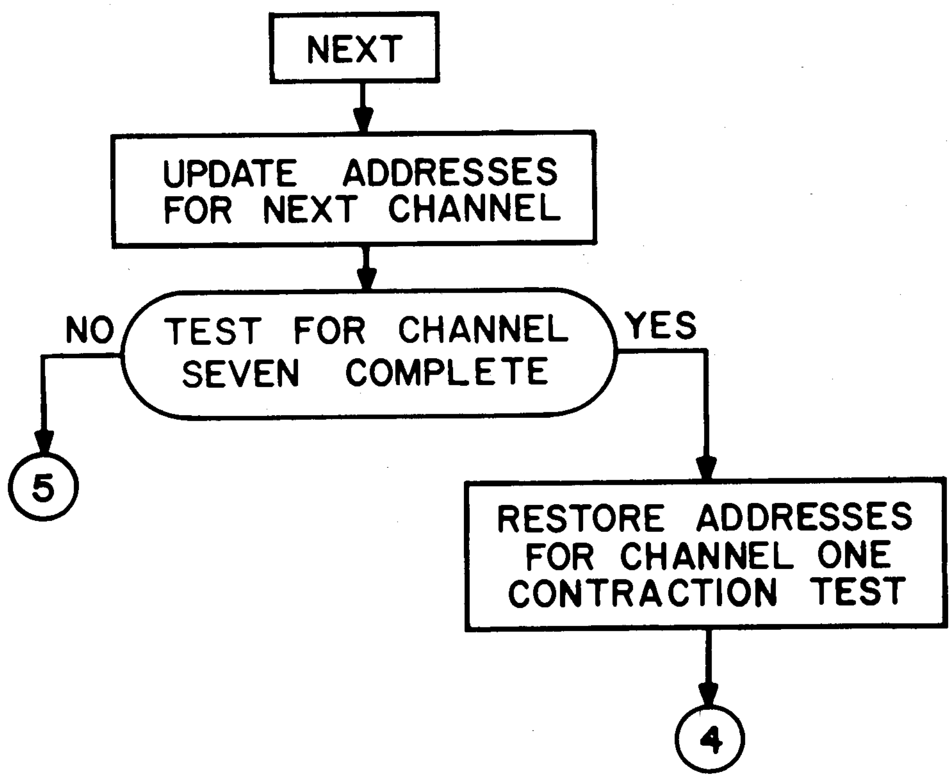


Figure 8. Flow chart of address modification section.

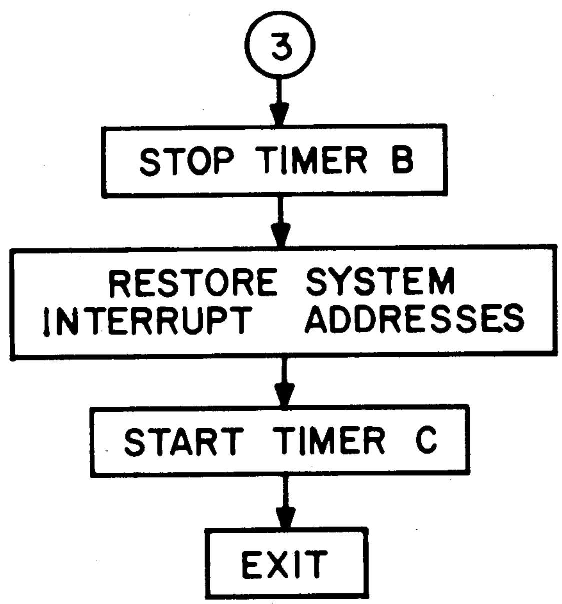


Figure 9. Flow chart of the exit routine.

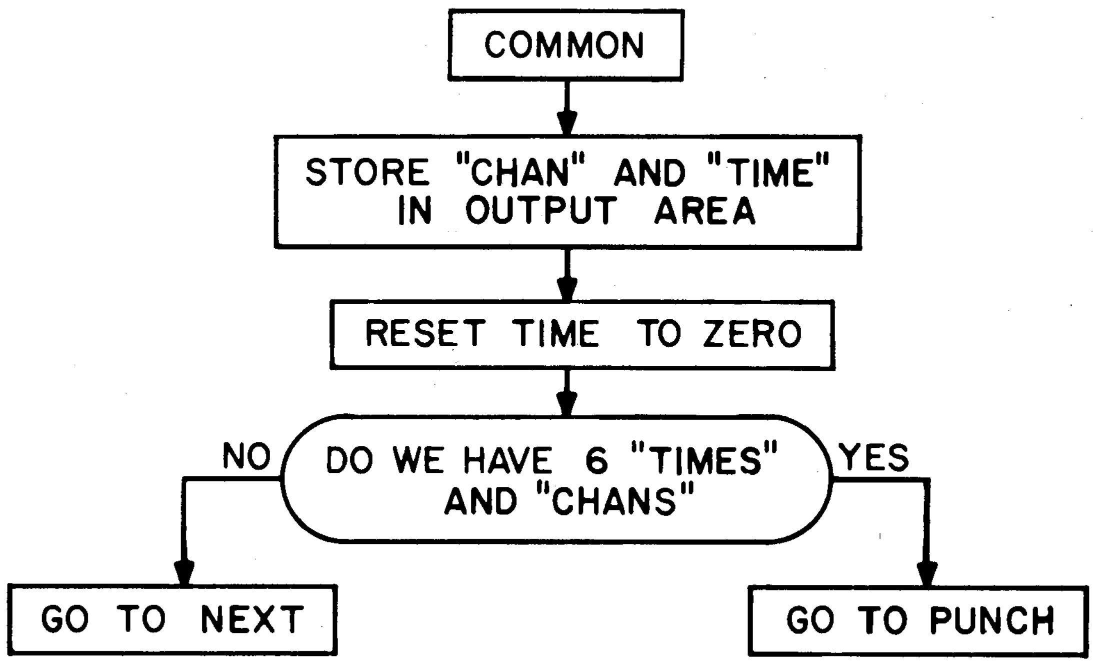


Figure 10. Flow chart of the common block.

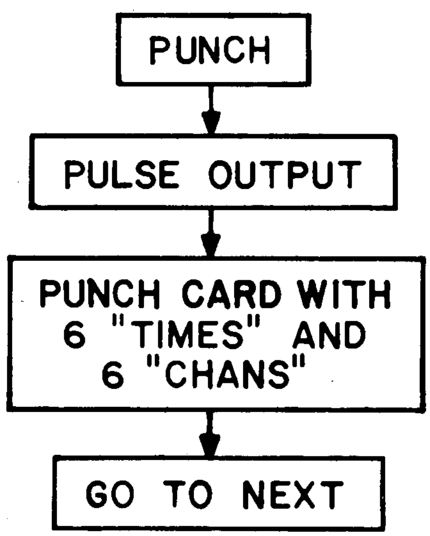


Figure 11. Flow chart of the punch routine.

```
                  ***************************************************
                  *READ MAG TAPE, SEVEN CHANNELS OR LESS SINGERMAN  *
                  *PUNCH TIME AND CHANNEL ID (SIX OF EACH PER CARD) *
                  *TIME IS TIME SINCE LAST CONTRACTION ON ANY       *
                  *CHANNEL.  CHANNEL IS CHANNEL I.D. OF CURRENT     *
                  *CONTRACTION.  CON3 IS BOTH THE SAMPLING RATE     *
                  *AND THE INCREMENT RATE OF "TIME"                 *
                  ***************************************************
0000 01 0C0001F2  START XIO  L  IOCCD       STOP TIMER C
0002 00 C400000B        LD   L  /000B
0004 01 D40001C3        STO  L  SAVEB
0006 00 2C40000B        STS  L  /000B,64    CLEAR S.P.B.
                  ***************************************************
                  *START                                            *
                  *STOP TIMER C                                     *
                  *SAVE SYSTEM TIMER INTERRUPT ADDRESS              *
                  *ESTABLISH PROGRAM INTERRUPT ADDRESSES            *
                  ***************************************************
0008 01 C400013A        LD   L  INTER
000A 00 D400000B        STO  L  /000B
                  ***************************************************
                  *CARD READER INITIALIZATION                       *
                  ***************************************************
000C 20 03059115        LIBF    CARDN       CARD FEED
000D 0  3000            DC      /3000
000E 0  0000            DC      0
000F 20 03059115        LIBF    CARDN       TEST CARD FEED
0010 0  0000            DC      0
0011 0  70FD            MDX     *-3
                  ***************************************************
                  *ENTRY OF CONSTANTS FOR DEFINING PROG. PARAMETERS *
                  ***************************************************
0012 20 23A17155        LIBF    TYPEN       WRITE MES1
0013 0  2002            DC      /2002
0014 1  0302            DC      DMES1-1
0015 0  0000            DC      0
0016 20 23A17155        LIBF    TYPEN       TEST WRITE COMPLETE
0017 0  0002            DC      /0002
0018 0  70FD            MDX     *-3
0019 20 17064885        LIBF    PAUSE
001A 1  01D6            DC      ONE
001B 01 0C0001E2        XIO  L  IOCC1       ENTER CON3 VIA D.E.S.
001D 20 23A17155        LIBF    TYPEN       WRITE MES2
001E 0  2002            DC      /2002
001F 1  031F            DC      DMES2-1
0020 0  0000            DC      0
0021 20 23A17155        LIBF    TYPEN       TEST WRITE COMPLETE
0022 0  0002            DC      /0002
0023 0  70FD            MDX     *-3
0024 20 17064885        LIBF    PAUSE
0025 1  01D7            DC      TWO
0026 01 0C0001E4        XIO  L  IOCC2       ENTER CON1 (1) VIA D.E.S.
0028 20 23A17155        LIBF    TYPEN       WRITE MES3
0029 0  2002            DC      /2002
002A 1  032E            DC      DMES3-1
002B 0  0000            DC      0
002C 20 23A17155        LIBF    TYPEN       TEST WRITE COMPLETE
002D 0  0002            DC      /0002
002E 0  70FD            MDX     *-3
002F 20 17064885        LIBF    PAUSE
0030 1  01D8            DC      THREE
0031 01 0C0001E6        XIO  L  IOCC3       ENTER CON1 (2) VIA D.E.S.
```

```
0033 20 23A17155          LIBF   TYPEN     WRITE MES4
0034 0  2002              DC     /2002
0035 1  033D              DC     DMES4-1
0036 0  0000              DC     0
0037 20 23A17155          LIBF   TYPEN     TEST WRITE COMPLETE
0038 0  0002              DC     /0002
0039 0  70FD              MDX    *-3
003A 20 17064885          LIBF   PAUSE
003B 1  01D9              DC     FOUR
003C 01 0C0001E8          XIO  L IOCC4     ENTER CON2 (1) VIA D.E.S.
003E 20 23A17155          LIBF   TYPEN     WRITE MES5
003F 0  2002              DC     /2002
0040 1  034C              DC     DMES5-1
0041 0  0000              DC     0
0042 20 23A17155          LIBF   TYPEN     TEST WRITE COMPLETE
0043 0  0002              DC     /0002
0044 0  70FD              MDX    *-3
0045 20 17064885          LIBF   PAUSE
0046 1  01DA              DC     FIVE
0047 01 0C0001EA          XIO  L IOCC5     ENTER CON2 (2) VIA D.E.S.
                    ***************************************************
                    *LISTING OF PROGRAM PARAMETERS DEFINED ABOVE      *
                    ***************************************************
0049 20 23A17140          LIBF   TYPE      WRITE CON3
004A 1  01D5              DC     CON3
004B 0  0001              DC     1
004C 20 040A3A17          LIBF   DBTYP     WRITE CON1
004D 1  0200              DC     CON1
004E 0  0001              DC     1
004F 20 040A3A17          LIBF   DBTYP     WRITE CON2
0050 1  0202              DC     CON2
0051 0  0001              DC     1
                    ***************************************************
                    *SET TIMER B TO SPECIFIED VALUE                   *
                    ***************************************************
0052 01 C40001D5          LD   L CON3
0054 00 D4000005          STO  L /0005
                    ***************************************************
                    *FINAL PROGRAM INITIALIZATION                     *
                    ***************************************************
0056 01 CC000212          LDD  L MIN
0058 0  61E4              LDX  1 -28
0059 01 DD000230    SETA  STD  L1 TEMP1+28
005B 0  7102              MDX  1 2
005C 0  70FC              MDX    SETA
005D 0  1010              SLA    16
005E 00 6500FF2E          LDX  L1 -210     SET UP 7 BLOCKS OF 30 CSL'S
0060 01 D5000302    SETB  STO  L1 BLOC1+210
0062 0  7101              MDX  1 1
0063 0  70FC              MDX    SETB
0064 01 D40003C4          STO  L AREAD+72
0066 01 D40003C5          STO  L AREAD+73
0068 01 D40001D4          STO  L CARDN
006A 20 23A17155          LIBF   TYPEN     WRITE MES6
006B 0  2002              DC     /2002
006C 1  035B              DC     DMES6-1
006D 0  0000              DC     0
006E 20 23A17155          LIBF   TYPEN     TEST WRITE COMPLETE
006F 0  0002              DC     /0002
0070 0  70FD              MDX    *-3
                    ***************************************************
                    *PAUSE TO START TAPE RECORDER                     *
```

```
                         ************************************************
0071 20 17064885                LIBF     PAUSE
0072 1  01DB                    DC       SIX
                         ************************************************
                         *START TIMER B                                 *
                         ************************************************
0073 01 0C0001F0                XIO  L   IOCCC       START TIMER B
                         ************************************************
                         *ESTABLISH LOOP FOR CONTRACTION TEST OF 7 CHANNELS*
                         ************************************************
0075 0  62F9                    LDX    2 -7
                         ************************************************
                         *WAIT 7                                        *
                         ************************************************
0076 0  3007             WAIT7 WAIT      7
                         ************************************************
                         *CONTRACTION TEST FOR ALL CHANNELS             *
                         ************************************************
0077 0  10A0                    SLT      32          SUM 30 CSL'S
0078 01 DC0001FE                STD  L   SUM
007A 0  61E2                    LDX    1 -30
007B 01 C500024E         ADD1   LD   L1  BLOC1+30
007D 0  1890                    SRT      16
007E 01 8C0001FE                AD   L   SUM
0080 01 DC0001FE                STD  L   SUM
0082 0  7101                    MDX    1 1
0083 0  70F7                    MDX      ADD1
0084 01 BC000214         CMPR1  DCM  L   TEMP1
0086 0  7009                    MDX      CONTC       SUM > TEMP1
0087 0  7001                    MDX      CONTD       SUM < TEMP1
0088 0  7000                    MDX      CONTD       SUM = TEMP1
0089 01 DC000214         CONTD  STD  L   TEMP1
008B 01 8C000200                AD   L   CON1
008D 01 DC000204         STOR1  STD  L   TEST1
008F 0  7047                    MDX      NEXT
0090 01 CC000212         CONTC  LDD  L   MIN
0092 01 DC000214         STOR2  STD  L   TEMP1
0094 01 CC0001FE                LDD  L   SUM
0096 01 BC000222         CMPR2  DCM  L   SMAX1
0098 0  7002                    MDX      CONTE       SUM > SMAX1
0099 0  7004                    MDX      CONTF       SUM < SMAX1
009A 0  7000                    MDX      CONTE       SUM = SMAX1
009B 01 DC000222         CONTE  STD  L   SMAX1
009D 0  7039                    MDX      NEXT      GO TO NEXT
009E 01 CC000222         CONTF  LDD  L   SMAX1
00A0 01 9C000202                SD   L   CON2
00A2 01 BC0001FE                DCM  L   SUM
00A4 0  7002                    MDX      CONTG       SMAX1-CON2 > SUM
00A5 0  7031                    MDX      NEXT      SMAX1-CON2 < SUM (TO NEXT)
00A6 0  7030                    MDX      NEXT      SMAX1-CON2 = SUM (TO NEXT)
00A7 01 CC000222         CONTG  LDD  L   SMAX1
00A9 01 BC000204         CMPR3  DCM  L   TEST1
00AB 0  700B                    MDX      CONTH       SMAX1 > TEST1
00AC 0  7001                    MDX      CONTI       SMAX1 < TEST1
00AD 0  7000                    MDX      CONTI       SMAX1 = TEST1
00AE 01 CC0001FE         CONTI  LDD  L   SUM
00B0 01 DC000214         STOR3  STD  L   TEMP1
00B2 01 CC000212                LDD  L   MIN
00B4 01 DC000222         STOR4  STD  L   SMAX1
00B6 0  7020                    MDX      NEXT      GO TO NEXT
00B7 01 C40001D6         CONTH  LD   L   ONE         STORE CHANNEL NO.
00B9 01 D40001D2                STO  L   CHAN
```

```
00BB 01 CC0001FE          LDD  L   SUM
00BD 01 DC000214    STOR5 STD  L   TEMP1
00BF 01 CC000212          LDD  L   MIN
00C1 01 DC000222    STOR6 STD  L   SMAX1
                    ****************************************************
                    *COMMON BLOCK (ALL "BLOC'S" COME HERE)             *
                    *STORE "TIME" AND "CHAN" IN OUTPUT AREA            *
                    ****************************************************
00C3 01 C40001C2    CONTJ LD   L   TIME        STORE TIME
00C5 01 D400036F    ADDRH STO  L   AREAC
00C7 01 C40001D2          LD   L   CHAN        STORE CHANNEL NO.
00C9 01 D4000370    ADDRI STO  L   AREAC+1
                    ****************************************************
                    *RESET "TIME" TO ZERO                              *
                    ****************************************************
00CB 0  1010              SLA      16
00CC 01 D40001C2          STO  L   TIME
00CE 01 740200C6          MDX  L   ADDRH+1,2   INCREMENT STORAGE & COUNTER
00D0 01 740200CA          MDX  L   ADDRI+1,2
                    ****************************************************
                    *DO WE HAVE 6 "TIMES" AND "CHANS"?                 *
                    ****************************************************
00D2 01 74FF01D3          MDX  L   CTNB,-1
00D4 0  7002              MDX      NEXT        RETURN
00D5 01 4C000119          BSC  L   PUNCH       GO TO PUNCH IF "6"
                    ****************************************************
                    *BEGIN NEXT                                        *
                    ****************************************************
00D7 01 741E007C    NEXT  MDX  L   ADD1+1,30
00D9 01 74020085          MDX  L   CMPR1+1,2
00DB 01 7402008A          MDX  L   CONTD+1,2
00DD 01 7402008E          MDX  L   STOR1+1,2
00DF 01 74020093          MDX  L   STOR2+1,2
00E1 01 74020097          MDX  L   CMPR2+1,2
00E3 01 7402009C          MDX  L   CONTE+1,2
00E5 01 7402009F          MDX  L   CONTF+1,2
00E7 01 740200A8          MDX  L   CONTG+1,2
00E9 01 740200AA          MDX  L   CMPR3+1,2
00EB 01 740200B1          MDX  L   STOR3+1,2
00ED 01 740200B5          MDX  L   STOR4+1,2
00EF 01 740100B8          MDX  L   CONTH+1,1
00F1 01 740200BE          MDX  L   STOR5+1,2
00F3 01 740200C2          MDX  L   STOR6+1,2
00F5 0  7201              MDX    2 1
00F6 0  7080              MDX      WAIT7+1
00F7 01 C4000118          LD   L   ADDRZ
00F9 0  D082              STO      ADD1+1
00FA 01 74F20085          MDX  L   CMPR1+1,-14
00FC 01 74F2008A          MDX  L   CONTD+1,-14
00FE 01 74F2008E          MDX  L   STOR1+1,-14
0100 01 74F20093          MDX  L   STOR2+1,-14
0102 01 74F20097          MDX  L   CMPR2+1,-14
0104 01 74F2009C          MDX  L   CONTE+1,-14
0106 01 74F2009F          MDX  L   CONTF+1,-14
0108 01 74F200A8          MDX  L   CONTG+1,-14
010A 01 74F200AA          MDX  L   CMPR3+1,-14
010C 01 74F200B1          MDX  L   STOR3+1,-14
010E 01 74F200B5          MDX  L   STOR4+1,-14
0110 01 74F900B8          MDX  L   CONTH+1,-7
0112 01 74F200BE          MDX  L   STOR5+1,-14
0114 01 74F200C2          MDX  L   STOR6+1,-14
0116 01 4C000075          BSC  L   WAIT7-1
```

```
                          ******************************************************
                          *END NEXT                                            *
                          *PUNCH                                               *
                          ******************************************************
0118 1  024E              ADDRZ DC        BLOC1+30
0119 01 74F400C6          PUNCH MDX   L   ADDRH+1,-12    PUNCH ROUTINE
011B 01 74F400CA                MDX   L   ADDRI+1,-12
011D 01 740601D3                MDX   L   CTUB,6
                          ******************************************************
                          *PULSE OUTPUT                                        *
                          ******************************************************
011F 01 0C0001F4                XIO   L   IOCCE         INITIATE PULSE
0121 01 0C0001F6                XIO   L   IOCCF         RESET PULSE
0123 0  61F4                    LDX    1  -12
0124 01 C500037B          LOOPA LD    L1  AREAC+12
0126 20 02255103                LIBF      BINDC
0127 1  037C              ADDRL DC        AREAD
0128 01 74060127                MDX   L   ADDRL,6
012A 0  7101                    MDX    1  1
012B 0  70F8                    MDX       LOOPA
012C 01 74B80127                MDX   L   ADDRL,-72
012E 01 740101D4                MDX   L   CARDN,1       INCREMENT CARD NO.
0130 01 C40001D4                LD    L   CARDN
0132 20 02255103                LIBF      BINDC
0133 1  03C6                    DC        AREAD+74
                          ******************************************************
                          *PUNCH CARD WITH 6 "TIMES" AND 6 "CHANS"             *
                          ******************************************************
0134 20 03059115                LIBF      CARDN
0135 0  2000                    DC        /2000
0136 1  037B                    DC        AREAD-1
0137 0  0000                    DC        0
0138 01 4C0000D7                BSC   L   NEXT         END OF PUNCH ROUTINE
                          ******************************************************
                          *TIMER INTERRUPT ROUTINE                             *
                          ******************************************************
013A 1  013B              INTER DC        TIMER
013B 0  0000              TIMER DC        *-*
013C 01 DC0001E0                STD   L   SAVEA         SAVE A AND Q
013E 00 CC000036                LDD   L   /0036
0140 01 DC0001FA                STD   L   SAVET         SAVE WK4 AND WK5
0142 01 C40001C4                LD    L   PROCD
0144 00 D4000037                STO   L   /0037
0146 01 2C0001AD                STS   L   HERE
0148 01 6D0001AF                STX   L1  XR1+1
014A 01 6E0001B1                STX   L2  XR2+1
014C 01 6F0001B3                STX   L3  XR3+1
014E 00 448000AD                BSI   I   173           GO TO TVEXIT
0150 01 DC0001FC          PROVE STD   L   ACCUM
0152 0  6914                    STX    1  XXR1+1
0153 0  6A15                    STX    2  XXR2+1
0154 0  2815                    STS       HHERE
0155 01 0C0001EC                XIO   L   IOCCA         READ TIMER D.S.W.
                          ******************************************************
                          *RESET TIMER B AND INCREMENT "TIME"                  *
                          ******************************************************
0157 01 C40001D5                LD    L   CON3           RESET TIMER B
0159 00 D4000005                STO   L   /0005
015B 01 740101C2                MDX   L   TIME,1        INCREMENT "TIME"
                          ******************************************************
                          *CLEAR INTERRUPT LEVEL                               *
                          ******************************************************
```

```
015D 01 4C40015F          BOSC L  CONTA
                    ****************************************************
                    *SAMPLE 7 ANALOG CHANNELS AND STORE                *
                    ****************************************************
015F 20 01262615    CONTA LIBF    AISQN       SAMPLE MULT.
0160 0  2200              DC      /2200
0161 1  01C5              DC      AREAA
0162 1  01CE              DC      ERROA
0163 20 01262615          LIBF    AISQN       TEST SAMPLE COMPLETE
0164 0  0000              DC      0
0165 0  70FD              MDX     *-3
0166 00 65000000    XXR1  LDX  L1 *-*
0168 00 66000000    XXR2  LDX  L2 *-*
016A 0  2000        WHERE LDS     *-*
016B 01 CC0001FC          LDD  L  ACCUM
016D 00 448000AC          BSI  I  172          GO TO TSAV
016F 0  C057              LD      AREAB       STORE SEVEN SAMPLES
0170 01 D4000230    ADDRA STO  L  BLOC1
0172 0  C055              LD      AREAB+1
0173 01 D400024E    ADDRB STO  L  BLOC2
0175 0  C053              LD      AREAB+2
0176 01 D400026C    ADDRC STO  L  BLOC3
0178 0  C051              LD      AREAB+3
0179 01 D400028A    ADDRD STO  L  BLOC4
017B 0  C04F              LD      AREAB+4
017C 01 D40002A8    ADDRE STO  L  BLOC5
017E 0  C04D              LD      AREAB+5
017F 01 D40002C6    ADDRF STO  L  BLOC6
0181 0  C04B              LD      AREAB+6
0182 01 D40002E4    ADDRG STO  L  BLOC7
                    ****************************************************
                    *UPDATE ADDRESSES FOR NEXT SAMPLING AND STORE OP   *
                    ****************************************************
0184 01 74010171          MDX  L  ADDRA+1,1   INCREMENT STORAGE LOCATIONS
0186 01 74010174          MDX  L  ADDRB+1,1
0188 01 74010177          MDX  L  ADDRC+1,1
018A 01 7401017A          MDX  L  ADDRD+1,1
018C 01 7401017D          MDX  L  ADDRE+1,1
018E 01 74010180          MDX  L  ADDRF+1,1
0190 01 74010183          MDX  L  ADDRG+1,1
                    ****************************************************
                    *TEST FOR 30 SETS OF SAMPLES                       *
                    ****************************************************
0192 01 74FF01D1          MDX  L  CNTA,-1
0194 0  7014              MDX     CONTB       AFTER 30 SAMPLES RESET
                    ****************************************************
                    *RESET ADDRESSES FOR SAMPLING AND STORE OPERATION  *
                    ****************************************************
0195 01 74E20171          MDX  L  ADDRA+1,-30       STORAGE LOCATIONS
0197 01 74E20174          MDX  L  ADDRB+1,-30
0199 01 74E20177          MDX  L  ADDRC+1,-30
019B 01 74E2017A          MDX  L  ADDRD+1,-30
019D 01 74E2017D          MDX  L  ADDRE+1,-30
019F 01 74E20180          MDX  L  ADDRF+1,-30
01A1 01 74E20183          MDX  L  ADDRG+1,-30
01A3 01 741E01D1          MDX  L  CNTA,30
                    ****************************************************
                    *TEST FOR EXIT                                     *
                    ****************************************************
01A5 0  0848              XIO     IOCCB       READ D.E.S.
01A6 0  E01A              AND     ETEST
01A7 0  4820              BSC     Z
```

```
01A8 0  700E                 MDX      STOP         STOP (IF NOT ZERO)
01A9 01 CC0001FA      CONTB  LDD   L  SAVET
01AB 00 DC000036             STD   L  /0036
01AD 0  2000          HERE   LDS      *-*
01AE 00 65000000      XR1    LDX   L1 *-*
01B0 00 66000000      XR2    LDX   L2 *-*
01B2 00 67000000      XR3    LDX   L3 *-*
01B4 0  C82B                 LDD      SAVEA
01B5 01 4C80013B             BSC   I  TIMER
                      ****************************************************
                      *EXIT ROUTINE                                      *
                      *STOP TIMER B                                      *
                      *RESTORE SYSTEM INTERRUPT ADDRESS                  *
                      *START TIMER C                                     *
                      ****************************************************
01B7 0  083A          STOP   XIO      IOCCD        STOP TIMER B
01B8 0  C00A                 LD       SAVEB
01B9 00 D400000B             STO   L  /000B
01BB 00 2C41000B             STS   L  /000B,65     SET S.P.B.
01BD 0  083A                 XIO      IOCCG        START TIMER C
01BE 0  C821                 LDD      SAVEA
01BF 30 059C98C0             EXIT
                      *      ONE-WORD CONSTANTS
01C1 0  4000          ETEST  DC       /4000
01C2 0  0000          TIME   DC       *-*
01C3 0  0000          SAVEB  DC       *-*
01C4 1  0150          PROCD  DC       PROVE
01C5 0  0008          AREAA  DC       8
01C6 0  100C                 DC       /100C
01C7    0007          AREAB  BSS      7
01CE 0  0000          ERROA  DC       *-*
01CF 01 4C8001CE             BSC   I  ERROA
01D1 0  001E          CNTA   DC       30
01D2 0  0000          CHAN   DC       *-*
01D3 0  0006          CTNB   DC       6
01D4 0  0000          CARDN  DC       0
01D5 0  0000          CON3   DC       *-*
01D6 0  0001          ONE    DC       1
01D7 0  0002          TWO    DC       2
01D8 0  0003          THREE  DC       3
01D9 0  0004          FOUR   DC       4
01DA 0  0005          FIVE   DC       5
01DB 0  0006          SIX    DC       6
01DC 0  0007          SEVEN  DC       7
01DD 0  1000          PULSE  DC       /1000
01DE 0  0000          RESET  DC       0
                      *      TWO-WORD CONSTANTS
01E0    0002          SAVEA  BSS   E  2
01E2 1  01D5          IOCC1  DC       CON3         ENTER CON3 VIA D.E.S.
01E3 0  0240                 DC       /0240
01E4 1  0200          IOCC2  DC       CON1         ENTER CON1 (1) VIA D.E.S.
01E5 0  0240                 DC       /0240
01E6 1  0201          IOCC3  DC       CON1+1       ENTER CON1 (2) VIA D.E.S.
01E7 0  0240                 DC       /0240
01E8 1  0202          IOCC4  DC       CON2         ENTER CON2 (1) VIA D.E.S.
01E9 0  0240                 DC       /0240
01EA 1  0203          IOCC5  DC       CON2+1       ENTER CON2 (2) VIA D.E.S.
01EB 0  0240                 DC       /0240
01EC 0  0000          IOCCA  DC       0            TIMER DSW
01ED 0  0721                 DC       /0721
01EE 0  0000          IOCCB  DC       0            READ D.E.S. INTO ACC.
01EF 0  0740                 DC       /0740
```

```
01F0 0  4000        IOCCC DC     /4000       START TIMER B
01F1 0  0420              DC     /0420
01F2 0  0000        IOCCD DC     0           STOP TIMERS
01F3 0  0420              DC     /0420
01F4 1  01DD        IOCCE DC     PULSE       SET E.C.O. (PULSE = /1000)
01F5 0  617F              DC     /617F
01F6 1  01DE        IOCCF DC     RESET       RESET E.C.O. (RESET = 0)
01F7 0  617F              DC     /617F
01F8 0  2000        IOCCG DC     /2000       START TIMER C
01F9 0  0420              DC     /0420
01FA    0002        SAVET BSS    2
01FC    0002        ACCUM BSS    2
01FE    0002        SUM   BSS    2
0200 0  0000        CON1  DC     *-*
0201 0  0000              DC     *-*
0202 0  0000        CON2  DC     *-*
0203 0  0000              DC     *-*
0204    0002        TEST1 BSS    2
0206    0002        TEST2 BSS    2
0208    0002        TEST3 BSS    2
020A    0002        TEST4 BSS    2
020C    0002        TEST5 BSS    2
020E    0002        TEST6 BSS    2
0210    0002        TEST7 BSS    2
0212 0  8000        MIN   DC     /8000
0213 0  0000              DC     /0000
0214    0002        TEMP1 BSS    2
0216    0002        TEMP2 BSS    2
0218    0002        TEMP3 BSS    2
021A    0002        TEMP4 BSS    2
021C    0002        TEMP5 BSS    2
021E    0002        TEMP6 BSS    2
0220    0002        TEMP7 BSS    2
0222    0002        SMAX1 BSS    2
0224    0002        SMAX2 BSS    2
0226    0002        SMAX3 BSS    2
0228    0002        SMAX4 BSS    2
022A    0002        SMAX5 BSS    2
022C    0002        SMAX6 BSS    2
022E    0002        SMAX7 BSS    2
0230    001E        BLOC1 BSS    30
024E    001E        BLOC2 BSS    30
026C    001E        BLOC3 BSS    30
028A    001E        BLOC4 BSS    30
02A8    001E        BLOC5 BSS    30
02C6    001E        BLOC6 BSS    30
02E4    001E        BLOC7 BSS    30
0302 0  001C              DC     DMESZ-DMES1
0303    0020        DMES1 DMES   '2RSEVEN: TO EXIT, SET D.E.S. 1.'E
0313    0018              DMES   'L'RENTER CON3 VIA D.E.S.'E
031F    0000        DMESZ BES    0
031F 0  000E              DC     DMESY-DMES2
0320    001C        DMES2 DMES   'L'RENTER CON1 (1) VIA D.E.S. 'E
032E    0000        DMESY BES    0
032E 0  000E              DC     DMESX-DMES3
032F    001C        DMES3 DMES   'L'RENTER CON1 (2) VIA D.E.S. 'E
033D    0000        DMESX BES    0
033D 0  000E              DC     DMESW-DMES4
033E    001C        DMES4 DMES   'L'RENTER CON2 (1) VIA D.E.S. 'E
034C    0000        DMESW BES    0
034C 0  000E              DC     DMESV-DMES5
034D    001C        DMES5 DMES   'L'RENTER CON2 (2) VIA D.E.S. 'E
```

```
035B      0000        DMESV BES     0
035B 0    0013              DC      DMESU-DMES6
035C      0020        DMES6 DMES    'L'RSTART TAPE RECORDER THEN PUSH '
036C      0006              DMES    START,'E
036F      0000        DMESU BES     0
036F      000C        AREAC BSS     12
037B 0    0050              DC      80
037C      0050        AREAD BSS     80
03CC      0000              END     START
```