

AD-A108 184

OKLAHOMA STATE UNIV STILLWATER ELECTRONICS LAB

F/G 9/2

AUTOMATED PCM ENCODER TEMPERATURE TEST, (U)

AUG 81 J W SPEARS

F19628-81-C-0079

UNCLASSIFIED

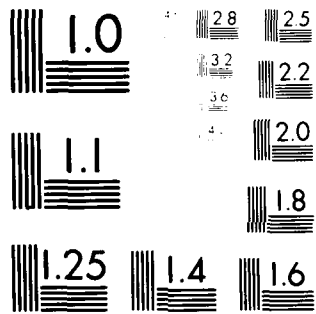
SCIENTIFIC-1

AFGL-TR-81-0242

NL

1 of 1
41A
D W 84

END
DATE
FILMED
01-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A108184

12 LEVEL II

AFGL-TR-81-0242

AUTOMATED PCM ENCODER TEMPERATURE TEST

J. W. Spears

Electronics Laboratory - D.E.T.A.
Office of Engineering Research
Oklahoma State University
Stillwater, Oklahoma 74078

15 August 1981

Scientific Report No. 1

DTIC
ELECTE
DEC 8 1981
S D
B

Approved for public release; distribution unlimited.

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

FILE COPY

81 12 08 232

Qualified requestors may obtain additional copies from the
Defense Technical Information Center. All others should
apply to the National Technical Information Service.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

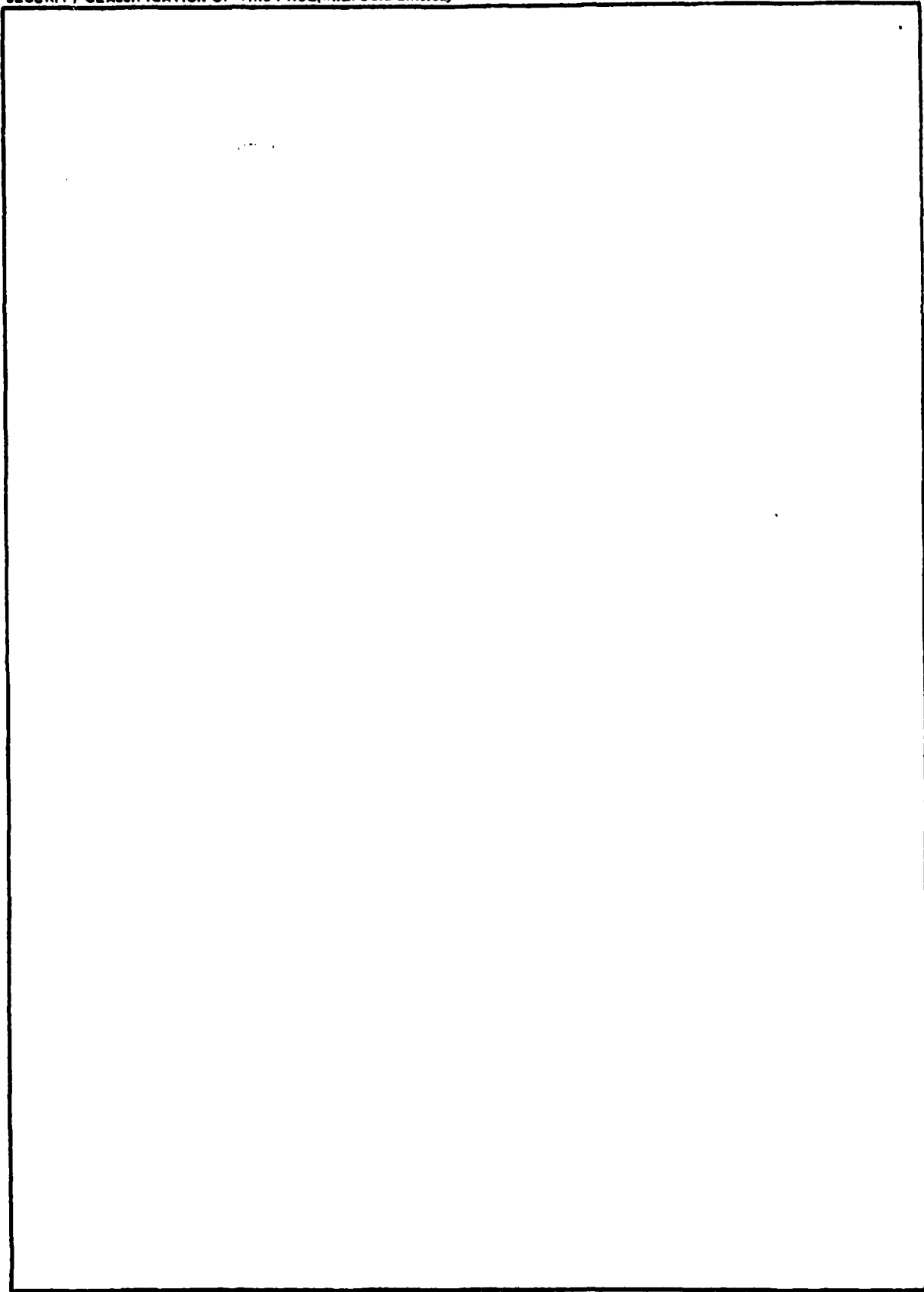
| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|--|-------------------------------------|--|
| 1. REPORT NUMBER AFGL-TR-81-0242 | 2. GOVT ACCESSION NO. AD-A108184 | 3. RECIPIENT'S CATALOG NUMBER 184 |
| 4. TITLE (and Subtitle) AUTOMATED PCM ENCODER TEMPERATURE TEST | | 5. TYPE OF REPORT & PERIOD COVERED Scientific Report No. 1 |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) J. W. SPEARS | | 8. CONTRACT OR GRANT NUMBER(s) F19628-81-C-0079 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Electronics Laboratory/- D.E.T.A. Oklahoma State University Stillwater, Oklahoma 74078 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 765904BB |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory Hanscom AFB, Massachusetts 01731 Monitor/Jack R. Griffin/LCR | | 12. REPORT DATE 15 August 1981 |
| | | 13. NUMBER OF PAGES 38 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | |
| 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) Testing, Automated routines, PCM Encoding, Microprocessors | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes an automated method for testing PCM encoder performance through use of microcomputer control and analysis of the test results. The microcomputer sequentially applies a known test input voltage to various inputs of the encoder under test, analyzes the digitally encoded signals from the encoder for a large number of samples, and provides a printout of the test results. A description of computer software and interface hardware necessary for the test is included. | | |

DD FORM 1473 1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified 4/14/81
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SUMMARY

Because of more stringent requirements for PCM encoder testing, an automated system was developed to evaluate and provide a permanent record of a PCM encoder's performance during temperature cycling. With this system, more extensive testing may be done in less time than required for the previous methods of encoder evaluation.

In the automated PCM encoder temperature test, a known voltage is applied to the input of the system. A microcomputer is used to sequentially apply this voltage to the various inputs of the encoder under test. The resultant digital output from the encoder is then fed to an associated PCM decommutator. The corresponding output signal from the PCM decommutator is then monitored by the microcomputer, which provides hardcopy output by printing the binary bit pattern, the calculated voltage, the error voltage (the difference between the calculated encoder output and input voltages), and the number of samples taken at each voltage level.

This report contains a description of the computer software and interface hardware necessary to set-up and operate the temperature test on a PCM encoder.

| | |
|---------------------|-------------------------------------|
| Accession For | |
| NTIS GRA&I | <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By _____ | |
| Distribution/ _____ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A | |

ACKNOWLEDGMENTS

The work discussed within this report was sponsored by the Aerospace Instrumentation Division of the Air Force Geophysics Laboratory, Hanscom Air Force Base, Massachusetts, under DOD contract F19628-78-C-0033 and, in the latest applications, under DOD contract F19628-81-C-0079.

Gratitude is expressed to the contract monitor, Jack R. Griffin/LCR, Sounding Rocket Branch, for the encouragement offered toward the work herein reported.

Thanks are also expressed to Joe Zinn, Bill Holloway, and Brian Cloer who were instrumental in the development of this automated test system.

TABLE OF CONTENTS
AUTOMATED PCM ENCODER TEMPERATURE TEST

| Topic | Page |
|---|------|
| 1.0 Introduction | 5 |
| 2.0 The Multiplex Breadboard | 5 |
| 3.0 KIM Interface Box | 8 |
| 4.0 Equipment for Test Setup | 8 |
| 5.0 Software | 10 |
| 5.1 Machine Code Algorithm ZIPMC | 10 |
| 5.1.1 Subroutine for Mainframe Data | 10 |
| 5.1.2 Variables Used in ZIPMC | 11 |
| 5.1.3 Subroutine for Subcom Data | 11 |
| 5.2 Basic Control Program | 13 |
| 5.2.1 Variables Used in ZIPB5 | 13 |
| 6.0 Running the Temperature Test | 15 |
| 7.0 Conclusions | 16 |
| APPENDIX A | |
| KIM Initialization For Temperature Test | 19 |
| APPENDIX B | |
| Software Listings | 22 |
| B.1 Basic Programs: | |
| B.1.1 ZIPB5 | 23 |
| B.1.2 IRBS4 | 26 |
| B.1.3 FIR1A | 28 |
| B.1.4 SPRD4 | 30 |
| B.2 Assembler Programs: | |
| B.2.1 ZIPA | 32 |
| B.2.2 ZIPAS | 34 |
| B.2.3 SPRDA | 36 |

LIST OF ILLUSTRATIONS

| Figure No. | | Page |
|------------|---|------|
| 1 | IRBS Temperature Test Block Diagram | 6 |
| 2 | FIRSSE Multiplex Breadboard | 7 |
| 3 | KIM Interface Box Schematic | 9 |
| 4 | Table for ZIPMC | 11 |
| 5 | Flowchart for ZIPMC | 12 |
| 6 | Example Printout of Temperature Test (FIRSSE) | 17 |
| 7 | Example Printout of Temperature Test (Spread F) | 18 |

1.0 Introduction

Once a new PCM encoder has been built, a normal requirement is environmental testing. The automated test described here will check the encoder's:

1. Prime data and subcom data inputs for wiring errors during construction.
2. Ability to convert analog to digital data within given specifications (normally $\pm \frac{1}{2}$ LSB).
3. Characteristics during temperature cycles (0° to 65°C). (There may be drift in the A/D in its full scale or zero offset, or most of all, an IC failure. If an IC failure occurs, the chip is replaced and the entire test repeated.)
4. Repeatability of sampled data.
5. Overall environmental noise level.

In the block diagram (Figure 1), the IRBS encoder is used as an example for the test set up. The only things that vary from testing one encoder to another are the "multiplex breadboard" and the software controlling the test.

The "multiplex breadboard" functions as a switching circuit, controlling a precision analog input voltage, set by a null-reading voltmeter. The switching circuit applies this voltage as an input signal to each encoder input line in a predetermined sequence, controlled by the KIM microcomputer through a 6-bit port on the KIM interface box. (This portion of the test set-up is a "breadboard" because the analog data inputs vary from encoder to encoder.)

The output of the encoder is fed into an OSU PCM decommutator, which provides digital output in parallel words and their associated addresses. These are monitored by the KIM through its interface box. Interaction between operator and computer is done through the CRT terminal. Once the test begins, it may be monitored on the CRT and a hard copy is provided by the printer.

A DAC may be used to monitor particular PCM words during the test.

2.0 The Multiplex Breadboard

The FIRSSE breadboard provides an example of a two link system and its "breadboard". HI-1818A chips are used here because of their low "on" resistance (approximately 200 ohms) and high isolation between channels (80 db). In Figure 2, the analog voltage is fed to all switching chips (IC 1-12) and the computer controls their address lines. These IC's are enabled by signals from two 3-to 8-line coders (IC's 13 & 14) and the "link enable" switch.

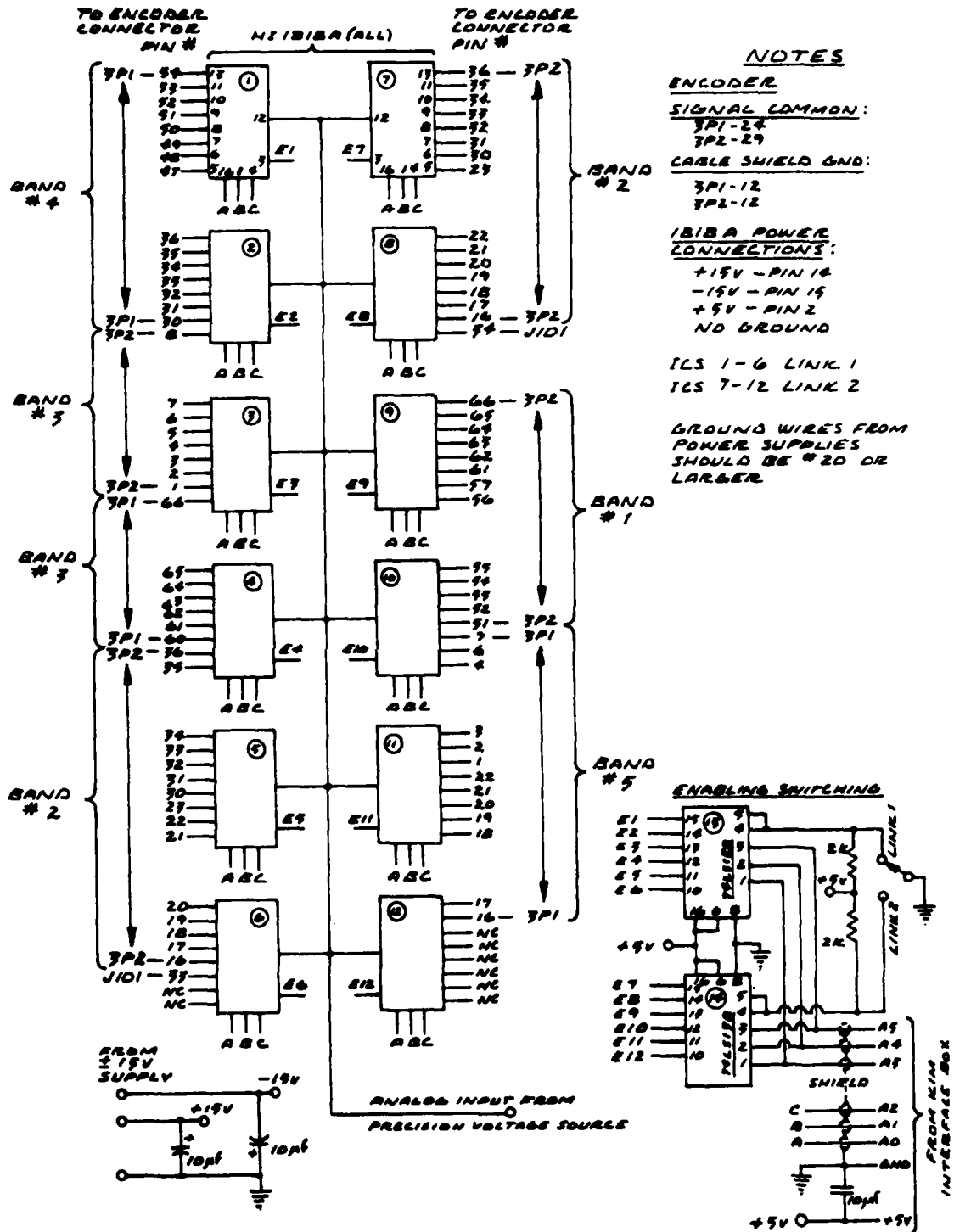


Figure 2. FIRSSE MULTIPLEX BREADBOARD

To help eliminate noise generated by the computer, it is essential to use bypass capacitors on all power supplies and shielded wire for the address lines from the computer.

The lowest address from the computer is 00. This should correspond to the first minor frame word to be tested (in the FIRSSE encoder this is word 03 on link 1 and word 08 on link 2). Then the address to the switching chips (from the computer) should increase sequentially in accord with the main frame word numbers on the encoder inputs. (Any alteration in this sequence creates more software changes.)

3.0 KIM Interface Box

This unit enables the computer to read the parallel data from the decom and to control the address lines to the "multiplex breadboard" which switches the input signal. (Schematic in Figure 3).

The word clock period from the decom is lengthened to 1.5u sec by the IC118 one-shot multivibrator. This is read by the computer through bit 7 of the 74LS253 multiplexers at address 0403.

The word address lines are then read at address 0400 through the same multiplexers. Once read, this address enables the data latches and the parallel data is latched in at addresses 0401 and 0402. This interface enables the computer to sample data at word rates up to 150 KHz.

Parallel data is also available at the buffered decom output for expansion to other devices. The parallel data and word address lines are buffered.

4.0 Equipment for Test Setup.

When an automated temperature test is performed at the Oklahoma State University Electronics Lab the following equipment is used:

Precision voltage source - EDC (RF-6146)

Precision voltmeter - Calibration Standards Corporation, Model DC-100A (RF-6145)

Appropriate "multiplex breadboard" for input signal switching.

Any OSU PCM decoder (Models D9ORP01, D9ORP21, or D9ORF01)

OSU KIM computer

Printer - Anadex DP-8000

CRT terminal - Hazeline 1500

KIM interface box

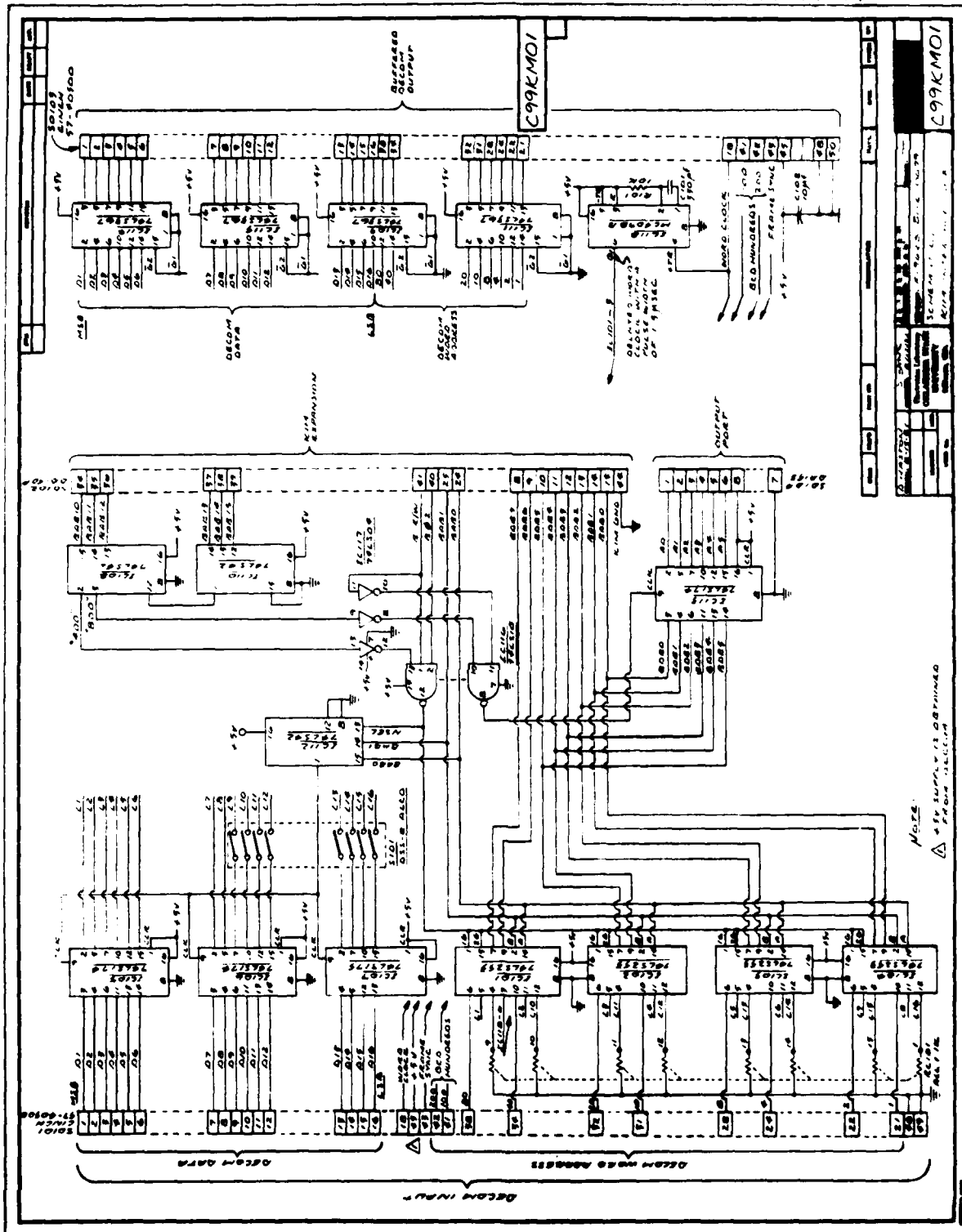


Figure 3. KIM INTERFACE BOX SCHEMATIC

Monitor DAC (optional): OSU 8-channel or DAC processor
Power supplies $\pm 15v$ and $+28v$
Encoder to be tested
OSU temperature test chamber

5.0 Software

The software used to test an encoder is a combination of machine code and BASIC programs. Together, these simplify obtaining data from the PCM decommutator. The machine code program is a general purpose routine that may be used for testing any PCM encoder. It requires the BASIC program to pass the output word address (for subframe data, to pass both the word and frame address) before calling it.

The BASIC program controls the format of the words to be sampled, operator data entry, and the data to the printer. This program is modified for each encoder to be tested.

5.1 Machine Code Program ZIPMC

There are two parts to this program: ZIPA for mainframe data (begins at address 7000) and ZIPAS for subcom data (begins at address 7100). These routines are identical, except the subcom routine monitors the subframe identification to obtain the subcom data. For the SPREAD F encoder, the program SPDMC uses SPRDA instead of ZIPAS for the subcom data selection.

5.1.1 Subroutine for Mainframe Data

This subroutine samples the specified word, storing binary bit patterns in array TABLE (Figure 4), counting the number of occurrences of each pattern. Each different binary pattern is retained in TABLE at a new cell address. If more than eight patterns are obtained, the data word is considered too noisy to test and the program terminates. Otherwise, the program terminates with 10,000 samples of the specified word. A flow chart of this routine is in Figure 5.

The total number of samples may be changed at address 7010 (high byte) and 7015 (low byte) and must be in hexadecimal numbers. For this routine, the number of samples may vary from 1 to 65,536 (2^{16}).

| Cell No. | Address | Binary Data | | Number of Samples | |
|----------|---------|-------------|-----|-------------------|-----|
| | | High | Low | High | Low |
| 1 | 7FC0 | | | | |
| 2 | 7FC4 | | | | |
| 3 | 7FC8 | | | | |
| 4 | 7FCC | | | | |
| 5 | 7FD0 | | | | |
| 6 | 7FD4 | | | | |
| 7 | 7FD8 | | | | |
| 8 | 7FDC | | | | |

Figure 4. TABLE FOR ZIPMC

5.1.2 Variables used in ZIPMC

X - Pointer to each address in TABLE.

Y - Cell pointer for next available address in TABLE to store binary data.

A - Obtains data from the interface box. The HEX addresses are:

0400 - decom word addresses.

0401 - decom data, bits 1-8 (MSBs), high binary data byte.

0402 - decom data, bits 9-16 (LSBs), low binary data byte.

0403 - decom word clock, positive pulse in bit 7.

TABLE - An array of 32 memory locations (addresses 7FC0 through 7FDF initialized to zero), divided into 8 cells of 4 locations each. The cells contain the decom data (high and low bytes) obtained from the accumulator (A), and the number of samples (high and low bytes) which is incremented when a sample is taken.

7FB0 - Up counter for low byte of number of the total samples. When equal to low byte of samples, this location is set to zero.

7FB1 - Location to save X so that X may be compared to Y.

7FB2, 7FB3 - Low and high bytes of the sample counter. Initialized to 10,000 (may be initialized to any number of samples), the low byte is compared to 7FB0 with each sample taken. When equal, the high address (7FB3) is decremented (if equal to zero the program terminates) and the low byte is set to FF.

5.1.3 Subroutine for Subcom Data

The algorithm, flow chart, and variables are the same for this subroutine

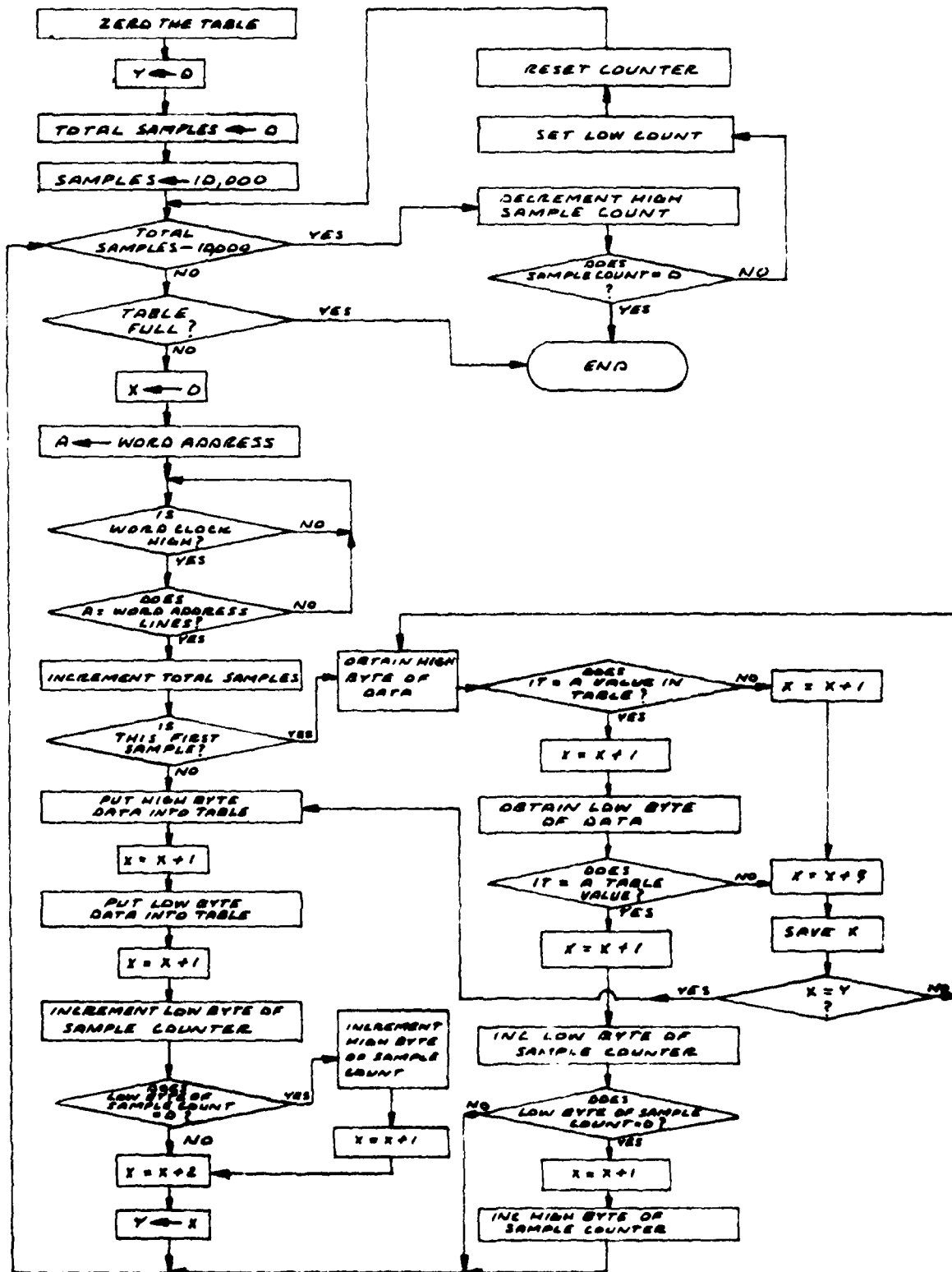


Figure 8. FLOW CHART FOR ZIPMC

as for the mainframe subroutine, except this routine obtains the subframe identification before searching for the decom word address.

The routine assumes the ID is in word one, but may be modified by changing address 712E (must use a hexadecimal number). The total number of samples is 100 (64 Hex) and may be changed at addresses 7115 (low byte) and 7110 (high byte). The program scans the subframe ID, shifts it right 2 bits and compares this to the frame number that BASIC stores at 714B. If an equal comparison, the program scans for the decom word address (BASIC stores this at 714F), if not equal; it continues to look for the correct ID.

For the various encoders that may be tested, the subframe ID may have to be shifted or bits may have to be blanked to obtain an ID in a format that can be compared to the desired frame number. The SPREAD F encoder subcom algorithm is an example of software shifting the ID so that a comparison may be made. This encoder's ID (word one) is in the most significant bits and is shifted right four bits.

5.2 BASIC Control Program

This routine provides interaction between the operator and computer, passes the word address and calls the machine code program, and then prints the results after the return from the call. This routine is not general purpose and must be modified for each encoder to be tested. Software listings for the ZIP, IRBS, FIRSSE, and SPREAD F encoders are in Appendix B.

The BASIC control program for ZIP is an example that will be discussed; the other programs in Appendix B are similar. The mainframe and subframe word addresses used are the main differences between the control programs.

For mainframe data, the ZIPB5 program samples words 12 through 43 (10,000 samples of each word) and samples selected subcom data from words 10 and 11 (100 samples of each subframe selected). In the program the variables are initialized, data is obtained from the operator, the machine code program is executed, and the data for each word sampled is printed.

5.3.1 Variables used in ZIPB5. (Program in Appendix B)

Numbers following variable are where variables initially occur. Section to control mainframe data (statements 29-115):

DT\$ - 29 Date of printout; should be manually changed to agree with current date. The date is printed in the title.

N - 45 An array that contains selected subframe numbers. The dimension is initialized to the total number of selected subframes. The selected subframes are read from the data statements by the FOR/NEXT loop (47, 48, 50).

ST,ED - 56 Starting and ending mainframe word addresses.

A - 71 Current word addresses.

HX - 80 Used in conversion of a BCD number to hexadecimal. In statement 100, HX is the converted number.

WA - 90 An intermediate variable used in conversion of a BCD number to hexadecimal.

8256, 8257 - 60, 65 Address locations to initialize the machine code subroutine starting address. 8256 is the low address (0) and 8257 is the high address (112 dec is 70 hex).

2048 - 72 Address location of KIM INTERFACE box output port. (2048 dec is 0800 hex).

28712 - 110 Address in machine code subroutine to store the current word address (28712 dec is 7000 hex).

Subroutine to call machine code program and print results (statements 119-340):

J,X1 - 119,120 The USR instruction requires this format but the variables are not used.

M - 130 TABLE index. Initialized to 32704 dec (7FC0 hex).

SM - 140 Low byte of the number of samples in TABLE.

SP - 153 High byte of the number of samples in TABLE.

B\$ - 170 String value to print the binary number in terms of 1's and 0's.

DA - 190 Obtains high and low bytes of the binary data to convert the data to a string value (B\$).

X - 195 Decimal value of each bit of the binary data.

FL - 270 Flag for loop indicator.

D1 - 290 High binary data byte.

D2 - 300 Low binary data byte.

V - 310 Calculated voltage from binary data.

VO - 311 Intermediate value for calculated voltage.

VD - 315 Voltage difference between the calculated and input voltages (error voltage).

Section to control subcom data (statements 350-610):

- WD - 405 Subcom word number.
- ST,ED - 410 Starting and ending loop values. Used to obtain a value from array N.
- K - 525 Current pointer to value in array N.
- FR - 530 Current subframe number.
- 29007 - 520 Address location to initialize the word address in the machine code program.
- 29003 - 570 Address location to store subframe number in machine code subroutine.

Other variables used in this section are similar to variables for the mainframe.

The list of selected subframe words is in the data statements 1000,1010. The first data statement is for subcom word 10 and the second for subcom word 11.

Subroutine to obtain information from operator and to print the Heading (1890-2100):

- TI\$ - 2000 Character string that is part of the title.
- LI - 2007 Link number.
- TP\$ - 2010 Temperature of chamber. May be in degrees F or C.
- VI - 2020 Input voltage (value from null reading volt meter).
- TS\$ - 2030 Used to move the output to the printer 5 spaces to the right.
- T\$ - 2040 Character string that is part of the title.
- DL - 2080 Kim interface output port variable. The port is assigned this value in statement 72.

6.0 Running the Temperature Test

To run the test the computer must be initialized (refer to KIM initialization in Appendix A), the equipment properly configured, and the PCM decommutator placed in the "all words" mode.

The operator must select an input voltage on the precision voltage source and null the voltmeter. If the computer has been initialized, the operator should type in the appropriate date (set DT\$ = "date"). Refer to the software listing for the statement numbers and format for changing DT\$. (In this mode any line typed in must be followed by the return key to enter this information.)

When the temperature chamber has reached the test temperature and remained there for 30 minutes (refer to "IRBS Acceptance Test Plan PCM Encoder Thermal Cycling" for procedures concerning thermal cycling), the operator may type in:

RUN. The computer will print out the title and date, and then asks the operator the link, temperature and the input voltage from the precision voltmeter. Refer to Figures 6 and 7 for example printouts.

The computer will then print out the word number, its binary value, its calculated voltage, the difference (error) between the calculated and the input voltage, and the number of samples at each voltage level. Up to 10,000 total samples are taken for each word (or only eight different voltage levels, whichever comes first).

If the subcom data is to be printed out (as in ZIP, SPREAD F, and IRBS), it follows the same procedures as the main frame data except the number of samples per word is reduced to 100 and the subcom word and frame numbers are written into the software.

7.0 Conclusions

The automated testing of PCM encoders has greatly improved the speed and quality of testing each unit. It allows more information to be gathered about an encoder's characteristics and performance.

Data was repeatable at given voltage levels. Even though the test set up actually introduced digital noise into the encoder during the test, the test is still considered valid because no more than ± 1 bit levels were observed (in most of the data only $\pm \frac{1}{2}$ bit was observed).

FIRSSE TEMPERATURE CHECKS 4/10/81

LINK? 2
 TEMPERATURE? 28 C
 INPUT VOLTAGE? 5.0105

| WORD# | BINARY VALUE | CALC VOLTAGE | ERROR | SAMPLES |
|-------|-------------------|--------------|-----------|---------|
| 8 | 11000000 00000000 | 5 | -.0105 | 9256 |
| 8 | 11000000 00010000 | 5.00488 | -5.62E-03 | 738 |
| 8 | 10111111 11110000 | 4.99511 | -.01539 | 6 |
| 9 | 10111111 11010000 | 4.98535 | -.02515 | 7410 |
| 9 | 10111111 11100000 | 4.99023 | -.02027 | 2585 |
| 9 | 10111111 11000000 | 4.98046 | -.03004 | 5 |
| 10 | 10111111 11010000 | 4.98535 | -.02515 | 8394 |
| 10 | 10111111 11000000 | 4.98046 | -.03004 | 1606 |
| 11 | 10111111 11100000 | 4.99023 | -.02027 | 6165 |
| 11 | 10111111 11110000 | 4.99511 | -.01539 | 3835 |
| 12 | 10111111 11100000 | 4.99023 | -.02027 | 9912 |
| 12 | 10111111 11010000 | 4.98535 | -.02515 | 63 |
| 12 | 10111111 11110000 | 4.99511 | -.01539 | 25 |
| 13 | 10111111 11100000 | 4.99023 | -.02027 | 8484 |
| 13 | 10111111 11010000 | 4.98535 | -.02515 | 1516 |
| 14 | 10111111 11010000 | 4.98535 | -.02515 | 9335 |
| 14 | 10111111 11100000 | 4.99023 | -.02027 | 660 |
| 14 | 10111111 11000000 | 4.98046 | -.03004 | 5 |
| 15 | 10111111 11100000 | 4.99023 | -.02027 | 7735 |
| 15 | 10111111 11110000 | 4.99511 | -.01539 | 2265 |
| 16 | 10111111 11100000 | 4.99023 | -.02027 | 9708 |
| 16 | 10111111 11110000 | 4.99511 | -.01539 | 269 |
| 16 | 10111111 11010000 | 4.98535 | -.02515 | 23 |
| 17 | 10111111 11110000 | 4.99511 | -.01539 | 613 |
| 17 | 10111111 11100000 | 4.99023 | -.02027 | 9385 |
| 17 | 10111111 11010000 | 4.98535 | -.02515 | 2 |
| 18 | 10111111 11100000 | 4.99023 | -.02027 | 7554 |
| 18 | 10111111 11010000 | 4.98535 | -.02515 | 2446 |
| 19 | 10111111 11110000 | 4.99511 | -.01539 | 9696 |
| 19 | 11000000 00000000 | 5 | -.0105 | 300 |
| 19 | 10111111 11100000 | 4.99023 | -.02027 | 4 |

Figure 6. EXAMPLE PRINTOUT OF TEMPERATURE TEST

SPREAD F ENCODER TEMPERATURE CHECKS 7/10/81

TEMPERATURE? OC
INPUT VOLTAGE? 4.499

| WORD# | BINARY VALUE | CALC VOLTAGE | ERROR | SAMPLES |
|-------|--------------|--------------|-----------|---------|
| 3 | 11100111 | 4.51171 | .0127 | 984 |
| 3 | 11100110 | 4.49218 | -6.83E-03 | 5 |
| 3 | 11101000 | 4.53125 | .03225 | 11 |
| 4 | 11100111 | 4.51171 | .0127 | 985 |
| 4 | 11101000 | 4.53125 | .03225 | 13 |
| 4 | 11100110 | 4.49218 | -6.83E-03 | 2 |
| 5 | 11100111 | 4.51171 | .0127 | 992 |
| 5 | 11100110 | 4.49218 | -6.83E-03 | 5 |
| 5 | 11101000 | 4.53125 | .03225 | 3 |
| 6 | 11100111 | 4.51171 | .0127 | 987 |
| 6 | 11101000 | 4.53125 | .03225 | 11 |
| 6 | 11100110 | 4.49218 | -6.83E-03 | 2 |
| 7 | 11100111 | 4.51171 | .0127 | 984 |
| 7 | 11101000 | 4.53125 | .03225 | 14 |
| 7 | 11100110 | 4.49218 | -6.83E-03 | 2 |

***** SUBFRAME DATA *****

WORD # 2
TEMPERATURE OC
INPUT VOLTAGE 4.499

| FRM# | BINARY VALUE | CALC VOLTAGE | ERROR | SAMPLES |
|------|--------------|--------------|-------|---------|
| 0 | 11100111 | 4.51171 | .0127 | 100 |
| 1 | 11100111 | 4.51171 | .0127 | 100 |
| 2 | 11100111 | 4.51171 | .0127 | 100 |
| 3 | 11100111 | 4.51171 | .0127 | 100 |
| 4 | 11100111 | 4.51171 | .0127 | 100 |
| 5 | 11100111 | 4.51171 | .0127 | 100 |
| 6 | 11100111 | 4.51171 | .0127 | 100 |

Figure 7. EXAMPLE PRINTOUT OF TEMPERATURE TEST

APPENDIX A

KIM INITIALIZATION FOR TEMPERATURE TEST

1. Turn on Hazeltine keyboard.
2. Turn on KIM computer.
 - Switch positions for computer
 - A. 1 MHz
 - B. Halt (Down)
 - C. Normal
 - D. Reset (Momentarily lift up)
 - E. Printer interface set to inhibit
3. Hit return on keyboard
 - A. Computer should respond with "KIM "
FEE6 FF
4. Type in: D400
5. Hit space bar:
 - A. Computer responds with D400 D8
6. Hit "G" key
7. KIM responds with "LMON"
"S"
8. Type in: MDC00, DCD0, 0000
9. Hit return
10. Hit return again (to get out of LMON mode)
 - A. KIM responds with "D400 D8"
11. Hit space bar
 - A. KIM responds with "0000 20"
12. Hit "G" key
13. Put in Disk #1B (FODS)
14. KIM responds "FODS"
15. Take out Disk #1B
16. Type in plus (+) sign (starts disk motor)
17. Put in Disk #10B
18. KIM responds #
19. Type in the appropriate machine code program from Table 1. Example:
LOD %ZIPMC (Hit Return)
20. KIM responds #
21. Take out the Disk
22. Type in + sign.

23. Put in Disk #1B, Basic. (For use with Anadex printer)
24. KIM responds "#"
25. Type in: RUN %BASP
26. Hit Return
27. KIM responds: Type in:

| | | |
|-----------------|-------|--------------|
| # Of Lines/Page | Ø | (Hit Return) |
| Memory Size | 286ØØ | (Hit Return) |
| Terminal Width | 8Ø | (Hit Return) |
28. Take disk out
29. Type in: Disk I (Hit return) (starts disk motor)
30. KIM responds: "OK"
31. Put in Disk (Refer to Table 1 for appropriate program from Disk 10B.)
32. Type in: DISKL, name
33. Hit Return
34. KIM should respond with "OK"
35. Take Disk out
36. Type in: QUIT (Hit Return)
37. KIM responds: "#"
38. Hit "ESC" key and KIM responds: "ØØØØ 4C"
39. On computer front panel do the following:
 - A. Move HALT to up position
 - B. 2 MHz
 - C. Move HALT to down position
 - D. Hit RESET
40. Hit return key on keyboard
41. KIM responds with "ØØØØ 4C"
42. Type in: 17F2 (Space Bar)
43. KIM should respond 17F2 ØB
44. If KIM responds 17F2 ØA then type in: ØB. (Make sure you type a period after ØB). This enters ØB into location 17F2 to set the baud rate between CRT and computer.
45. Hit the Space Bar
46. Hit "G" key
47. KIM should respond "OK."
48. Type in: RUN (Hit return to start program.) If the printer is to be used, set printer interface switch to "HANDSHAKE".

TABLE 1 Programs are on Disk 10B.

| Encoder under Test | Program Name | Machine Code Program |
|-----------------------|--------------|-------------------------|
| ZIP | ZIPB5 | ZIPMC |
| IRBS | IRBS4 | ZIPMC |
| FIRSSE | FIR1A | ZIPMC |
| SPREAD F | SPRD4 | SPDMC |

APPENDIX B
SOFTWARE LISTINGS

B.1 BASIC PROGRAMS:

- B.1.1 ZIPB5
- B.1.2 IRBS4
- B.1.3 FIR1A
- B.1.4 SPRD4

B.2 ASSEMBLER PROGRAMS:

- B.2.1 ZIPA
- B.2.2 ZIPAS
- B.2.3 SPRDA

```

15 REM      ZIPB5.....ZIP-II ENCODER TEMPERATURE TEST
16 REM
20 REM      THIS PROGRAM TAKES 10000 SAMPLES OF EACH MAINFRAME WORD
21 REM      AND 100 SAMPLES OF EACH SUBCOM WORD.  WORDS 12 THRU 43
22 REM      ARE SAMPLED FIRST THEN THE SUBFRAME WORDS.
23 REM      THIS ROUTINE CALLS THE MACHINE CODE PROGRAM "ZIPMC"
24 REM      TO OBTAIN MAINFRAME AND SUBFRAME DATA.
25 REM
28 PRINT
29 DT$="1/27/80"
30 PRINT TAB(22);"ZIP-II  TEMPERATURE TEST  ";DT$
35 PRINT : PRINT:
40 PRINT
43 REM      ....INITIALIZE ARRAY N FOR SUBFRAME DATA.....
45 DIM N(25) : REM      ...FOR THESE CHECKS, 25 SELECTED SUBCOM
46 REM      WORDS WILL BE PRINTED.
47 FOR I=0 TO 24
48 READ N(I)
50 NEXT I
55 GOSUB 2000 : REM      ....GET DATA FROM OPERATOR.....
56 ST=12 : ED=43 : REM.....SET STARTING & ENDING MF WORD #'S
58 REM      ....INITIALIZE STARTING ADDR FOR ZIPMC
59 REM      THIS WILL READ MAINFRAME DATA
60 POKE 8256,0
65 POKE 8257,112
70 REM      ....BEGIN LOOP FOR MAINFRAME DATA.....
71 FOR A=ST TO ED
72 POKE 2048,DL
73 DL=DL+1
80 HX=INT(A/10)
90 WA=A-(HX*10)
100 HX=HX*16+WA
110 POKE 28712,HX
111 GOSUB 119
112 PRINT : NEXT A
115 GOTO 380
117 REM      ....SUBROUTINE TO CALL MACHINE CODE PROGRAM
118 REM      AND PRINT RESULTS. (STATEMENTS 119-340)
119 J=0
120 X1=USR(J)
130 M=32704
140 SM=PEEK(M+2)
153 SP=PEEK(M+3)
155 IF SP+SM=0 THEN RETURN
160 FL=0
170 B$=""
190 DA=PEEK(M)
195 X=128
200 IF X<=DA THEN 230
210 B$=B$+"0"
220 GOTO 240
230 B$=B$+"1"
235 DA=DA-X
240 X=X/2
250 IF X>.5 THEN GOTO 200
260 DA=PEEK(M+1)
270 FL=FL+1
275 B$=B$+" "

```

```

280 IF FL=1 THEN GOTO 195
290 D1=PEEK(M)
300 D2=PEEK(M+1)
310 V=(D1+D2/256)/12.8
311 VD=V
312 SM=(SP*256)+SM
313 V=INT((VD-10)*100000)/100000
315 VD=V-VI
316 VD=INT(VD*100000)/100000
320 PRINT TS$;A;TAB(12);B$;TAB(34);V;TAB(50);VD;TAB(66);SM
330 M=M+4
340 GOTO 140
345 REM
350 REM      ....ROUTINE TO OBTAIN AND PRINT SUBCOM DATA....
355 REM
380 PRINT
400 PRINT
402 PRINT TS$;"***** SUBFRAME DATA *****"
405 WD=10
410 ST=0 ; ED=15
415 FRINT : PRINT
420 PRINT TS$;"LINK ";LI
435 PRINT TS$;"WORD # ";WD
436 PRINT TS$;"TEMPERATURE ";TP$
437 PRINT TS$+"INPUT VOLTAGE ";VI
440 PRINT
450 R$="FRM#      BINARY VALUE      CALC VOLTAGE      ERROR      "
460 PRINT TS$+R$+T1$
478 POKE 8256,0
480 POKE 8257,113
490 HX=INT(WD/10)
500 WA=WD-(HX*10)
510 HX=HX*16+WA
520 POKE 29007,HX
525 FOR K=ST TO ED
530 FR=N(K)
535 PRINT
550 HX=INT(FR/10)
555 WA=FR-(HX*10)
560 HX=HX*16+WA
570 POKE 29003,HX
575 A=FR
580 GOSUB 119
590 NEXT K
595 IF K>24 THEN END
600 ST=16 : ED=24
610 WD=11: GOTO 415
900 REM
910 REM      ....LIST OF SUBFRAME WORDS.  FIRST DATA
920 REM      STATEMENT IS FOR SUBCOM WORD 10.  THE
930 REM      SECOND IS FOR WORD 11.
1000 DATA 0,8,13,16,24,32,37,41,48,55,56,64,69,72,80,85
1010 DATA 16,24,32,40,60,68,72,80,87
1890 REM
1900 REM      ....SUBROUTINE TO OBTAIN LINK,TEMPERATURE, &
1910 REM      INFUT VOLTAGE FROM OPERATOR AND TO
1920 REM      PRINT DATA HEADING.
1930 REM
2000 T1$="      SAMPLES"
2005 PRINT

```

```
2007 INPUT"      LINK";LI
2010 INPUT "      TEMPERATURE";TP$
2020 INPUT "      INPUT VOLTAGE";VI
2030 TS$="      "
2040 T$="WORD*      BINARY VALUE      CALC VOLTAGE      ERROR      "
2050 PRINT
2060 PRINT TS$+T$+T1$
2070 PRINT
2075 REM      ....INITIALIZE DL (THE OUTPUT FORT VARIABLE)
2080 IF LI=1 THEN DL=0
2090 IF LI=2 THEN DL=32
2100 RETURN
OK
```

```

5 REM      IRBS4.....IRBS ENCODER TEMPERATURE TEST
6 REM
10 REM     THIS ROUTINE TAKES 10000 SAMPLES OF THE IRBS ENCODER
20 REM     ANALOG INPUTS FOR THE MAINFRAME DATA.
25 REM
26 REM     THIS ROUTINE CALLS ZIPMC TO OBTAIN DATA FROM THE
27 REM     DECOM. IT USES BOTH THE MAINFRAME AND SUBFRAME
28 REM     ROUTINES OF ZIPMC. THE 'DATA' STATEMENTS DEFINE
29 REM     THE SELECTED SUBCOM WORDS.
30 PRINT ; PRINT
35 DT$="7/10/80"
37 PRINT TAB(24);"IRBS TEMPERATURE CHECKS   ";DT$
38 PRINT ; PRINT
45 DL=0
46 PRINT
48 T1$="      SAMPLES"
50 INPUT "TEMPERATURE";TP$
51 INPUT "INPUT VOLTAGE";VI
52 ST=5 : ED=32
53 PRINT
55 T$="WORD#      BINARY VALUE          CALC VOLTAGE          ERROR   "
56 PRINT T$+T1$
57 PRINT
58 FOR A=ST TO ED
60 POKE 8256,0
70 POKE 8257,112
72 POKE 2048,DL
73 DL=DL+1
80 HX=INT(A/10)
90 WA=A-(HX*10)
100 HX=HX*16+WA
110 POKE 28712,HX
111 GOSUB 119
112 GOTO 350
119 J=0
120 X1=USR(J)
130 M=32704
140 SM=PEEK(M+2)
153 SP=PEEK(M+3)
155 IF SP+SM=0 THEN RETURN
160 FL=0
170 B$=""
190 DA=PEEK(M)
195 X=128
200 IF X<=DA THEN 230
210 B$=B$+"0"
220 GOTO 240
230 B$=B$+"1"
235 DA=DA-X
240 X=X/2
250 IF X).5 THEN GOTO 200
260 DA=PEEK(M+1)
270 FL=FL+1
275 B$=B$+" "
280 IF FL=1 THEN GOTO 195
290 D1=PEEK(M)
300 D2=PEEK(M+1)
310 V=(D1+D2/256)/51.2

```

```

311 VD=V
312 SM=(SP*256)+SM
313 V=INT((VD)*100000)/100000
315 VD=V-VI
316 VD=INT(VD*100000)/100000
320 PRINT TAB(0);A;TAB(7);B$;TAB(29);V;TAB(45);VD;TAB(61);SM
330 M=M+4
340 GOTO 140
350 PRINT
360 NEXT A
370 IF ST=35 THEN GOTO 380
371 ST=35 : ED=37
372 GOTO 58
380 PRINT
400 PRINT
402 PRINT "***** SUBCOM DATA *****"
403 PRINT
404 DIM N(20)
405 WD=33
432 OR I=1 TO 20
433 READ N(I)
434 NEXT I
435 PRINT "WORD # ";WD
436 PRINT "TEMPERATURE ";TP$
437 PRINT "INPUT VOLTAGE ";VI
440 R1$="          SAMPLES"
450 R$="FRM*      BINARY VALUE      CALC VOLTAGE      ERROR      "
460 PRINT;PRINT R$+R1$
478 POKE 8256,0
480 POKE 8257,113
490 HX=INT(WD/10)
500 WA=WD-(HX*10)
510 HX=HX*16+WA
518 F=1
519 MX=10
520 POKE 29007,HX
525 IF WD=33 THEN GOTO 530
526 F=11
527 MX=17
530 FOR K=F TO MX
532 FR=N(K)
535 PRINT
540 HX=FR
570 POKE 29003,HX
575 A=FR
580 GOSUB 119
590 NEXT K
595 PRINT
596 PRINT
599 IF WD=34 THEN END
600 DL=0
609 WD=34
610 GOTO 435
700 DATA 2,9,18,26,34,42,49,58,66,74
710 DATA 5,13,21,29,37,45,53,58,66,74
OK

```

LIST

```

20 REM:  FIR1...ROUTINE TO CHECK OUT THE FIRSPE ENCODER.  USED IN
21 REM:      CONJUNCTION WITH ZIPMC, THIS ROUTINE PRINTS OUT THE
22 REM:      CALCULATED VOLTAGE VALUES, BINARY VALUE OF THE WORD
23 REM:      BEING SAMPLED, THE DIFFERENCE BETWEEN THE INPUT
24 REM:      AND CALCULATED VOLTAGES, AND THE NUMBER OF SAMPLES
25 REM:      AT THE BINARY VALUE.  ONLY 8 DIFFERENT BINARY
26 REM:      VALUES ARE STORED AND PRINTED FOR EACH WORD SAMPLED
27 REM:      THEN THE PROGRAM ENDS (DOESN'T PROCEED TO NEXT WORD).
28 REM:
29 REM:      THE DECOM MUST BE IN ALL WORDS POSITION AND THE
30 REM:      INTERFACE BOX SET FOR 12 BITS/WORD.
31 REM:
32 REM:      VARIABLES:
33 REM:      DL = ADDRESS COUNTER TO MUX BOARD
44 GOSUB 1000 : REM PRINT HEADING
45 DL=0
47 ED=49
48 T1$="      SAMPLES"
49 INPUT "      LINK ";LI
50 INPUT "      TEMPERATURE ";TP$
51 INPUT "      INPUT VOLTAGE ";VI
52 IF LI=1 THEN ST=3
53 PRINT : PRINT
54 IF LI=2 THEN ST=8
55 T$="WORD#      BINARY VALUE          CALC VOLTAGE          ERROR      "
56 PRINT TS$+T$+T1$
57 PRINT
58 FOR A=ST TO ED
59 IF A=3 THEN DL=45
60 POKE 8256,0
70 POKE 8257,112
72 POKE 2048,DL
73 IF A=3 THEN DL=-1
75 DL=DL+1
80 HX=INT(A/10)
90 WA=A-(HX*10)
100 HX=HX*16+WA
110 POKE 28712,HX
111 GOSUB 119
112 GOTO 350
119 J=0
120 X1=USR(J)
130 M=32704
140 SM=PEEK(M+2)
153 SP=PEEK(M+3)
155 IF SP+SM=0 THEN RETURN
160 FL=0
170 B$=""
190 DA=PEEK(M)
195 X=128
200 IF X<=DA THEN 230
210 B$=B$+"0"
220 GOTO 240
230 B$=B$+"1"
235 DA=DA-X
240 X=X/2
250 IF X>.5 THEN GOTO 200
260 DA=PEEK(M+1)

```

```

270 FL=FL+1
275 B$=B$+" "
280 IF FL=1 THEN GOTO 195
290 D1=PEEK(M)
300 D2=PEEK(M+1)
310 V=(D1+D2/256)/12.8
311 VD=V
312 SM=(SP*256)+SM
313 V=INT((VD-10)*100000)/100000
315 VD=V-VI
316 VD=INT(VD*100000)/100000
320 PRINT TAB(6);A;TAB(12);B$;TAB(34);V;TAB(50);VD;TAB(66);SM
330 M=M+4
340 GOTO 140
350 PRINT
355 IF A=3 THEN A=4
360 NEXT A
370 POKE 2048,00
440 IF LI=2 THEN END
450 PRINT
550 INPUT "      IS LINK 2 READY (Y=YES)";Y$
560 IF Y$="Y" THEN GOTO 590
570 END
590 IF LI=1 THEN LI=2
595 PRINT ; PRINT
598 PRINT "      LINK ?"; LI
600 PRINT "      TEMPERATURE ?"; TP$
605 PRINT "      INPUT VOLTAGE ?"; VI
608 DL=0
610 GOTO 52
1000 DT$="4/10/81"
1010 TS$="      "
1020 PRINT ; PRINT ; PRINT
1030 PRINT TAB(20);"FIR SSE ENCODER TEMPERATURE TEST ";DT$
1040 PRINT ; PRINT ; PRINT
1050 RETURN

```

OK

```

15 REM   SPRD4.....SPREAD F ENCODER TEMPERATURE TEST
16 REM
20 REM   THIS PROGRAM TAKES 1000 SAMPLES OF EACH MAINFRAME WORD
21 REM   AND 100 SAMPLES OF EACH SUBCOM WORD.  WORDS 3 THRU 7
22 REM   ARE SAMPLED FIRST THEN THE SUBFRAME WORDS.
23 REM   THIS ROUTINE CALLS THE MACHINE CODE PROGRAM "SPDMC"
24 REM   TO OBTAIN MAINFRAME AND SUBFRAME DATA.
25 REM
26 REM   THIS PROGRAM DOES NOT PRINT THE PARITY BIT (BIT 9).
27 REM
28 PRINT
29 DT$="7/8/81"
30 PRINT TAB(18);"SPREAD F ENCODER TEMPERATURE CHECKS   ";DT$: PRINT : FRIN
40 PRINT
42 DL=8
43 REM   ....INITIALIZE ARRAY N FOR SUBFRAME DATA.....
45 DIM N(16)
47 FOR I=0 TO 15
48 READ N(I)
50 NEXT I
55 GOSUB 2000 : REM   ....GET DATA FROM OPERATOR.....
56 ST=3 : ED=7 : REM.....SET STARTING & ENDING MF WORD *
58 REM   ....INITIALIZE STARTING ADDR FOR SPDMC.....
59 REM   THIS WILL READ MAINFRAME DATA
60 POKE 8256,0
65 POKE 8257,112
70 REM   ....BEGIN LOOP FOR MAINFRAME DATA.....
71 FOR A=ST TO ED
72 POKE 2048,DL
73 DL=DL+1
80 HX=INT(A/10)
90 WA=A-(HX*10)
100 HX=HX*16+WA
110 POKE 28712,HX
111 GOSUB 119
112 PRINT : NEXT A
115 GOTO 380
117 REM   ....SUBROUTINE TO CALL MACHINE CODE PROGRAM
118 REM   AND PRINT RESULTS. (STATEMENTS 119-340)
119 J=0
120 X1=USR(J) : REM   ....CALL MACHINE CODE PROGRAM....
130 M=32704
140 SM=PEEK(M+2)
153 SP=PEEK(M+3)
155 IF SP+SM=0 THEN RETURN
160 FL=0
170 B$=""
190 DA=PEEK(M)
195 X=128
200 IF X(=DA THEN 230
210 B$=B$+"0"
220 GOTO 240
230 B$=B$+"1"
235 DA=DA-X
240 X=X/2
250 IF X).5 THEN GOTO 200
260 DA=PEEK(M+1)
290 D1=PEEK(M)

```

```

300 D2=PEEK(M+1)
310 V=(D1+D2/256)/51.2
311 VD=V
312 SM=(SP*256)+SM
313 V=INT((VD)*100000)/100000
315 VD=V-VI
316 VD=INT(VD*100000)/100000
320 PRINT TS$;A;TAB(16);B$;TAB(34);V;TAB(50);VD;TAB(66);SM
330 M=M+4
340 GOTO 140
345 REM
350 REM      ....ROUTINE TO OBTAIN AND PRINT SUBCOM DATA....
355 REM
380 PRINT
400 PRINT
402 PRINT TS$;"***** SUBFRAME DATA *****"
403 PRINT
405 WD=2
435 PRINT TS$;"WORD * ";WD
436 PRINT TS$;"TEMPERATURE ";TP$
437 PRINT TS$;"INPUT VOLTAGE ";VI
450 R$="FRM*      BINARY VALUE      CALC VOLTAGE      ERROR      "
460 PRINT : PRINT TS$+R$+T1$
478 POKE 8256,0
480 POKE 8257,113
490 HX=INT(WD/10)
500 WA=WD-(HX*10)
510 HX=HX*16+WA
520 POKE 29000,HX
525 FOR FR=0 TO 15
530 DL=N(FR)
535 PRINT
545 POKE 2048,DL
570 POKE 28996,FR
575 A=FR
580 GOSUB 119
590 NEXT FR
600 DL=8
610 END
900 REM
910 REM      ....DATA TO CONTROL MUX BREADBOARD
920 REM      TO OBTAIN SUBFRAME DATA....
930 REM
1000 DATA 13,14,15,16,13,17,18,19,13,14,15,16,13,20,21,22
1890 REM
1900 REM      ....SUBROUTINE TO OBTAIN TEMPERATURE AND
1910 REM      INPUT VOLTAGE FROM OPERATOR AND TO
1920 REM      PRINT DATA HEADING.
1930 REM
2000 T1$="      SAMPLES"
2005 PRINT
2010 INPUT "      TEMPERATURE";TP$
2020 INPUT "      INPUT VOLTAGE";VI
2030 TS$="      "
2040 T$="WORD*      BINARY VALUE      CALC VOLTAGE      ERROR      "
2050 PRINT
2060 PRINT TS$+T$+T1$
2070 PRINT
2080 RETURN

```

OK

```

10 0000      ; ZIFA.....FOR MAIN FRAME DATA
20 0000      ;
30 0000      ; 6/30/80
40 0000      ;
50 0000      ; ZIPMC IS THE MACHINE CODE FORM OF THIS PROGRAM.
60 0000      ;
70 0000      ;
80 0000      ; SUBROUTINE TO BE USED IN CONJUNCTION WITH BASIC
90 0000      ; TO LOOK AT DATA FROM THE DECOM....USE WITH ZIPB..
100 0000     ;
110 0000     ; THIS ROUTINE SAMPLES A WORD 10,000 TIMES AND
120 0000     ; COUNTS THE * OF WORDS THAT ARE A LIKE.  THERE
130 0000     ; ARE ONLY 8 DIFFERENT BIT PATTERNS RETAINED.
140 0000     ;
150 7000     *=$7000
160 7000     TABLE =$7FC0
170 7000     ;
180 7000 A240 ZIFA   LDX #$40   ; ZERO THE TABLE
190 7002 A900     LDA #0
200 7004 9DBF7F LOOPA  STA $7FBF,X
210 7007 CA       DEX
220 7008 D0FA     BNE LOOPA
230 700A A000     LDY #0
240 700C 8CB07F   STY $7FB0   ; TOTAL SAMPLES=0
250 700F A927     LDA #$27   ; SET HI BYTE OF * OF SAMPLES
260 7011 8DB37F   STA $7FB3
270 7014 A937     LDA #$37   ; SET LOW BYTE OF * OF SAMPLES
280 7016 8DB27F   STA $7FB2
290 7019 ADB27F   TSAM   LDA $7FB2   ; GET * OF SAMPLES
300 701C CDB07F   CMP $7FB0
310 701F F066     BEQ CKSAM
320 7021 C020     NOISE  CPY #$20   ; IF=20 THEN "ITS REAL NOISY!!"
330 7023 F067     BEQ END
340 7025 A200     TABPT  LDX #$00   ; SET TABLE POINTER=0
350 7027 A900     LDA #$00   ; GET WORD * FROM BASIC
360 7029 2C0304   WDCLK  BIT $403   ; IS WORD CLOCK HI?
370 702C 10FB     BPL WDCLK
380 702E CD0004   LOOPB  CMP $400   ; GET WORD ADDRESS
390 7031 D0F6     ENE WDCLK
400 7033 EEB07F   INC $7FB0   ; INC THE TOTAL SAMPLES
410 7036 C000     CPY #00    ; IS THIS FIRST SAMPLE
420 7038 D01E     BNE NEWDAT
430 703A AD0104   GETHI  LDA $401   ; GET HI DATA (MSB'S)
440 703D 9DC07F   STA TABLE,X ; PUT INTO TABLE
450 7040 EB       INX
460 7041 AD0204   LDA $402   ; GET LOW DATA
470 7044 9DC07F   STA TABLE,X
480 7047 EB       INX
490 7048 FEC07F   INC TABLE,X ; INC* SAMPLES FOR THIS DATA
500 704B D005     BNE XPLUS  ; IF LD CNT=0 THEN INC HI CNT
510 704D EB       INX
520 704E FEC07F   INC TABLE,X ; INC HI CNT
530 7051 CA       DEX
540 7052 EB       XPLUS  INX
550 7053 EB       INX
560 7054 BA       TXA     ; SET Y=X
570 7055 AB       TAY
580 7056 10C1     BPL TSAM   ; GO CHECK TOTAL SAMPLES

```

```

590 7058 AD0104 NEWDAT LDA $401 ; GET HI DATA
600 705B DDC07F CMP TABLE,X ; = TO TABLE VALUE?
610 705E FO0E BEQ TESTLO
620 7060 E8 SETX3 INX ; X=X+4
630 7061 E8 SETX2 INX
640 7062 E8 INX
650 7063 E8 INX
660 7064 8EB17F STX $7FB1 ; SAVE X TO CMP
670 7067 CCB17F CPY $7FB1 ; DOES X=Y??
680 706A DOEC BNE NEWDAT ; GO GET NEXT TABLE VALUE
690 706C FOCC BEQ GETHI ; GO PUT INTO TABLE
700 706E E8 TESTLO INX
710 706F AD0204 LDA $402 ; GET LOW DATA
720 7072 DDC07F CMP TABLE,X ; = TO TABLE VALUE?
730 7075 DOEA BNE SETX2
740 7077 E8 INX
750 7078 FEC07F INC TABLE,X ; INC THE SAMPLE CNT
760 707B FO03 BEQ HISAM ; IF=0 THEN INC HI SAMPLE CNT
770 707D 4C1970 JMP TSAM
780 7080 E8 HISAM INX
790 7081 FEC07F INC TABLE,X ; INC HI SAM CNT
800 7084 4C1970 JMP TSAM
810 7087 CEB37F CKSAM DEC $7FB3 ; DEC HI SAMPLE CNT
820 708A 1001 BPL SETLO
830 708C 60 END RTS
840 708D A9FF SETLO LDA #$FF ; SET LO CNT
850 708F 8DB27F STA $7FB2
860 7092 A900 LDA #$00 ; RESET CNT
870 7094 8DB07F STA $7FB0
880 7097 4C1970 JMP TSAM

```

```

10 0000      ; ZIPAS.....ROUTINE TO OBTAIN SUBCOM DATA
20 0000      ;
30 0000      ; 6/30/80
40 0000      ;
50 0000      ; SUBROUTINE TO BE USED IN CONJUNCTION WITH BASIC
60 0000      ; TO LOOK AT DATA FROM THE DECOM....USE WITH ZIPB..
70 0000      ;
80 0000      ; THIS ROUTINE SAMPLES A SUBCOM WORD 100 TIMES AND
90 0000      ; COUNTS THE # OF WORDS THAT ARE A LIKE.  THERE
100 0000     ; ARE ONLY 8 DIFFERENT BIT PATTERNS RETAINED.
110 0000     ;
120 7100     *=$7100
130 7100     TABLE =$7FC0
140 7100     ;
150 7100 A240 ZIPA   LDX #$40   ; ZERO THE TABLE
160 7102 A900     LDA #0
170 7104 9DBF7F LOOPA  STA $7FBF,X
180 7107 CA       DEX
190 7108 DOFA     BNE LOOPA
200 710A A000     LDY #0
210 710C 8CB07F   STY $7FB0   ; TOTAL SAMPLES=0
220 710F A900     LDA #$00
230 7111 8DB37F   STA $7FB3
240 7114 A964     LDA #$64   ; SET SAMPLES=100
250 7116 8DB27F   STA $7FB2
260 7119 ADB27F   TSAM   LDA $7FB2   ; GET # OF SAMPLES
270 711C CDB07F   CMP $7FB0
280 711F D003     BNE NOISE
290 7121 4CAE71   JMP CKSAM
300 7124 C020     NOISE  CPY #$20   ; IF=20 THEN "ITS REAL NOISY!!"
310 7126 D003     BNE TABPT
320 7128 4CB371   JMP END
330 712B A200     TABPT  LDX #$00   ; SET TABLE POINTER=0
340 712D A901     LOOP1  LDA #$01   ; LOOK FOR ID
350 712F 2C0304   WDCLK  BIT $403   ; IS WORDD CLOCK HI?
360 7132 10FB     BPL WDCLK
370 7134 CD0004   CMP $400   ; GET WORDD ADDRESS
380 7137 DOF6     BNE WDCLK
390 7139 AD0104   LDA $401   ; GET HI DATA
400 713C 8DB47F   STA $7FB4   ; SAVE
410 713F AD0204   LDA $402   ; GET LO DATA
420 7142 6EB47F   ROR $7FB4
430 7145 6A       ROR A      ; SHIFT 2 MSB OF 10 INTO A
440 7146 6EB47F   ROR $7FB4
450 7149 6A       ROR A
460 714A C900     CMP #0      ; CMP WITH DESIRED FRAME#
470 714C D0DF     BNE LOOP1   ; GO LOOK FOR CORRECT FRAME
480 714E A900     LDA #0      ; GET WORD* FROM BASIC
490 7150 2C0304   WDCLK1  BIT $403   ; IS WORD CLK HI?
500 7153 10FB     BPL WDCLK1
510 7155 CD0004   CMP $400   ; GET WORD ADDRESS
520 7158 DOF6     BNE WDCLK1
530 715A EEB07F   INC $7FB0   ; INC THE TOTAL SAMPLES
540 715D C000     CPY #0      ; IS THIS FIRST SAMPLE?
550 715F D01E     BNE NEWDAT
560 7161 AD0104   GETHI  LDA $401   ; GET HI DATA
570 7164 9DC07F   STA TABLE,X ; PUT INTO TABLE
580 7167 E8       INX

```

| | | | | | |
|------|------|--------|--------|-------------|-------------------------------|
| 590 | 7168 | AD0204 | | LDA \$402 | ; GET LOW DATA |
| 600 | 716B | 9DC07F | | STA TABLE,X | |
| 610 | 716E | E8 | | INX | |
| 620 | 716F | FEC07F | | INC TABLE,X | ; INC# SAMPLES FOR THIS DATA |
| 630 | 7172 | D005 | | BNE XPLUS | ; IF LO CNT=0 THEN INC HI CNT |
| 640 | 7174 | E8 | | INX | |
| 650 | 7175 | FEC07F | | INC TABLE,X | ; INC HI CNT |
| 660 | 7178 | CA | | DEX | |
| 670 | 7179 | E8 | XPLUS | INX | |
| 680 | 717A | E8 | | INX | |
| 690 | 717B | 8A | | TXA | ; SET Y=X |
| 700 | 717C | A8 | | TAY | |
| 710 | 717D | 109A | | BPL TSAM | ; GO CHECK TOTAL SAMPLES |
| 720 | 717F | AD0104 | NEWDAT | LDA \$401 | ; GET HI DATA |
| 730 | 7182 | DDC07F | | CMP TABLE,X | ; = TO TABLE VALUE? |
| 740 | 7185 | FOOE | | BEQ TESTLO | |
| 750 | 7187 | E8 | SETX3 | INX | ; X=X+4 |
| 760 | 7188 | E8 | SETX2 | INX | |
| 770 | 7189 | E8 | | INX | |
| 780 | 718A | E8 | | INX | |
| 790 | 718B | 8EB17F | | STX \$7FB1 | ; SAVE X TO CMP |
| 800 | 718E | CCB17F | | CPY \$7FB1 | ; DOES X=Y?? |
| 810 | 7191 | DOEC | | BNE NEWDAT | ; GO GET NEXT TABLE VALUE |
| 820 | 7193 | FOCC | | BEQ GETHI | ; GO PUT INTO TABLE |
| 830 | 7195 | E8 | TESTLO | INX | |
| 840 | 7196 | AD0204 | | LDA \$402 | ; GET LOW DATA |
| 850 | 7199 | DDC07F | | CMP TABLE,X | ; = TO TABLE VALUE? |
| 860 | 719C | DOEA | | BNE SETX2 | |
| 870 | 719E | E8 | | INX | |
| 880 | 719F | FEC07F | | INC TABLE,X | ; INC THE SAMPLE CNT |
| 890 | 71A2 | F003 | | BEQ HISAM | ; IF=0 THEN INC HI SAMPLE CNT |
| 900 | 71A4 | 4C1971 | | JMP TSAM | |
| 910 | 71A7 | E8 | HISAM | INX | |
| 920 | 71AB | FEC07F | | INC TABLE,X | ; INC HI SAM CNT |
| 930 | 71AB | 4C1971 | | JMP TSAM | |
| 940 | 71AE | CEB37F | CKSAM | DEC \$7FB3 | ; DEC HI SAMPLE CNT |
| 950 | 71B1 | 1001 | | BPL SETLO | |
| 960 | 71B3 | 60 | END | RTS | |
| 970 | 71B4 | A9FF | SETLO | LDA #\$FF | ; SET LO CNT |
| 980 | 71B6 | 8DB27F | | STA \$7FB2 | |
| 990 | 71B9 | A900 | | LDA #\$00 | ; RESET CNT |
| 1000 | 71BB | 8DB07F | | STA \$7FB0 | |
| 1010 | 71BE | 4C1971 | | JMP TSAM | |

```

10 0000      ; SPRDA.....SPREAD F ENCODER SUBCOM CHECKOUT
20 0000      ;
30 0000      ; 7/9/81
40 0000      ;
50 0000      ; SUBROUTINE TO BE USED IN CONJUNCTION WITH BASIC
60 0000      ; TO LOOK AT DATA FROM DECOM....USE WITH SPRDF
70 0000      ;
80 0000      ; THIS ROUTINE SAMPLES A SUBCOM WORD 100 TIMES AND
90 0000      ; COUNTS THE * OF WORDS THAT ARE A LIKE. THERE
100 0000     ; ARE ONLY 8 DIFFERENT BIT PATTERNS RETAINED.
110 0000     ;
120 7100     *=$7100
130 7100     TABLE =$7FC0
140 7100
150 7100 A240 ZIPA   LDX #$40      ; ZERO THE TABLE
160 7102 A900      LDA #0
170 7104 9DBF7F LOOPA  STA $7FBF,X
180 7107 CA        DEX
190 7108 DOFA      BNE LOOPA
200 710A A000      LDY #0
210 710C 8CB07F    STY $7FB0      ; TOTAL SAMPLES=0
220 710F A900      LDA #$00
230 7111 8DB37F    STA $7FB3
240 7114 A964      LDA #$64      ; SET SAMPLES=100
250 7116 8DB27F    STA $7FB2
260 7119 ADB27F    TSAM   LDA $7FB2      ; GET # OF SAMPLES
270 711C CDB07F    CMP $7FB0
280 711F D003      BNE NOISE
290 7121 4CA771    JMP CKSAM
300 7124 C020      NOISE  CPY #$20      ; IF=20 THEN "ITS REAL NOISY!!"
310 7126 D003      BNE TABPT
320 7128 4CAC71    JMP END
330 712B A200      TABPT  LDX #$00      ; SET TABLE POINTER=0
340 712D A901      LOOP1  LDA #$01      ; LOOK FOR ID
350 712F 2C0304    WDCLK  BIT $403      ; IS WORDD CLOCK HI?
360 7132 10FB      BPL WDCLK
370 7134 CD0004    CMP $400      ; GET WORDD ADDRESS
380 7137 DOF6      BNE WDCLK
390 7139 AD0104    LDA $401      ; GET HI DATA
400 713C 8DB47F    STA $7FB4      ; SAVE
410 713F 6A        ROR A         ; SHIFT ID RIGHT 4 BITS
420 7140 6A        ROR A
430 7141 6A        ROR A
440 7142 6A        ROR A
450 7143 C900      CMP #0        ; CMP WITH DESIRED FRAME#
460 7145 DOE6      BNE LOOP1     ; GO LOOK FOR CORRECT FRAME
470 7147 A900      LDA #0        ; GET WORD* FROM BASIC
480 7149 2C0304    WDCLK1  BIT $403      ; IS WORD CLK HI?
490 714C 10FB      BPL WDCLK1
500 714E CD0004    CMP $400      ; GET WORD ADDRESS
510 7151 DOF6      BNE WDCLK1
520 7153 EEB07F    INC $7FB0     ; INC THE TOTAL SAMPLES
530 7156 C000      CPY #0        ; IS THIS FIRST SAMPLE?
540 7158 D01E      BNE NEWDIAT
550 715A AD0104    GETHI  LDA $401      ; GET HI DATA
560 715D 9DC07F    STA TABLE,X ; PUT INTO TABLE
570 7160 EB        INX
580 7161 AD0204    LDA $402      ; GET LOW DATA

```

| | | | | | | |
|------|------|--------|--------|-----|------------|-----------------------------|
| 590 | 7164 | 9DC07F | | STA | TABLE, X | |
| 600 | 7167 | EB | | INX | | |
| 610 | 7168 | FEC07F | | INC | TABLE, X ; | INC* SAMPLES FOR THIS DATA |
| 620 | 716B | D005 | | BNE | XPLUS ; | IF LO CNT=0 THEN INC HI CNT |
| 630 | 716D | EB | | INX | | |
| 640 | 716E | FEC07F | | INC | TABLE, X ; | INC HI CNT |
| 650 | 7171 | CA | | DEX | | |
| 660 | 7172 | EB | XPLUS | INX | | |
| 670 | 7173 | EB | | INX | | |
| 680 | 7174 | 8A | | TXA | | ; SET Y=X |
| 690 | 7175 | A8 | | TAY | | |
| 700 | 7176 | 10A1 | | BPL | TSAM ; | GO CHECK TOTAL SAMPLES |
| 710 | 7178 | AD0104 | NEWDAT | LDA | \$401 ; | GET HI DATA |
| 720 | 717B | DDC07F | | CMP | TABLE, X ; | = TO TABLE VALUE? |
| 730 | 717E | FOOE | | BEQ | TESTLO | |
| 740 | 7180 | EB | SETX3 | INX | | ; X=X+4 |
| 750 | 7181 | EB | SETX2 | INX | | |
| 760 | 7182 | EB | | INX | | |
| 770 | 7183 | EB | | INX | | |
| 780 | 7184 | 8EB17F | | STX | \$7FB1 ; | SAVE X TO CMP |
| 790 | 7187 | CCB17F | | CPY | \$7FB1 ; | DOES X=Y?? |
| 800 | 718A | DOEC | | BNE | NEWDAT ; | GO GET NEXT TABLE VALUE |
| 810 | 718C | FOCC | | BEQ | GETHI ; | GO PUT INTO TABLE |
| 820 | 718E | EB | TESTLO | INX | | |
| 830 | 718F | AD0204 | | LDA | \$402 ; | GET LOW DATA |
| 840 | 7192 | DDC07F | | CMP | TABLE, X ; | = TO TABLE VALUE? |
| 850 | 7195 | DOEA | | BNE | SETX2 | |
| 860 | 7197 | EB | | INX | | |
| 870 | 7198 | FEC07F | | INC | TABLE, X ; | INC THE SAMPLE CNT |
| 880 | 719B | F003 | | BEQ | HISAM ; | IF=0 THEN INC HI SAMPLE CNT |
| 890 | 719D | 4C1971 | | JMP | TSAM | |
| 900 | 71A0 | EB | HISAM | INX | | |
| 910 | 71A1 | FEC07F | | INC | TABLE, X ; | INC HI SAM CNT |
| 920 | 71A4 | 4C1971 | | JMP | TSAM | |
| 930 | 71A7 | CEB37F | CKSAM | DEC | \$7FB3 ; | DEC HI SAMPLE CNT |
| 940 | 71AA | 1001 | | BPL | SETLO | |
| 950 | 71AC | 60 | END | RTS | | |
| 960 | 71AD | A9FF | SETLO | LDA | ##FF ; | SET LO CNT |
| 970 | 71AF | 8DB27F | | STA | \$7FB2 | |
| 980 | 71B2 | A900 | | LDA | ##00 ; | RESET CNT |
| 990 | 71B4 | 8DB07F | | STA | \$7FB0 | |
| 1000 | 71B7 | 4C1971 | | JMP | TSAM | |

END

DATE
FILME

01-8

DTIC